ABSTRACT

Title of Dissertation: A PREDICTIVE MODEL OF NUCLEAR POWER PLANT CREW DECISION-MAKING AND PERFORMANCE IN A DYNAMIC SIMULATION ENVIRONMENT

Kevin A. Coyne, Doctor of Philosophy, 2009

Dissertation directed by: Professor Ali Mosleh, Department of Mechanical Engineering

The safe operation of complex systems such as nuclear power plants requires close coordination between the human operators and plant systems. In order to maintain an adequate level of safety following an accident or other off-normal event, the operators often are called upon to perform complex tasks during dynamic situations with incomplete information. The safety of such complex systems can be greatly improved if the conditions that could lead operators to make poor decisions and commit erroneous actions during these situations can be predicted and mitigated. The primary goal of this research project was the development and validation of a cognitive model capable of simulating nuclear plant operator decision-making during accident conditions.

Dynamic probabilistic risk assessment methods can improve the prediction of human error events by providing rich contextual information and an explicit consideration of feedback arising from man-machine interactions. The Accident

Dynamics Simulator paired with the Information, Decision, and Action in a Crew context cognitive model (ADS-IDAC) shows promise for predicting situational contexts that might lead to human error events, particularly knowledge driven errors of commission. ADS-IDAC generates a discrete dynamic event tree (DDET) by applying simple branching rules that reflect variations in crew responses to plant events and system status changes. Branches can be generated to simulate slow or fast procedure execution speed, skipping of procedure steps, reliance on memorized information, activation of mental beliefs, variations in control inputs, and equipment failures. Complex operator mental models of plant behavior that guide crew actions can be represented within the ADS-IDAC mental belief framework and used to identify situational contexts that may lead to human error events.

This research increased the capabilities of ADS-IDAC in several key areas. The ADS-IDAC computer code was improved to support additional branching events and provide a better representation of the IDAC cognitive model. An operator decision-making engine capable of responding to dynamic changes in situational context was implemented. The IDAC human performance model was fully integrated with a detailed nuclear plant model in order to realistically simulate plant accident scenarios. Finally, the improved ADS-IDAC model was calibrated, validated, and updated using actual nuclear plant crew performance data. This research led to the following general conclusions:

- A relatively small number of branching rules are capable of efficiently capturing a wide spectrum of crew-to-crew variabilities.

- Compared to traditional static risk assessment methods, ADS-IDAC can provide a more realistic and integrated assessment of human error events by directly determining the effect of operator behaviors on plant thermal hydraulic parameters.
- The ADS-IDAC approach provides an efficient framework for capturing actual operator performance data such as timing of operator actions, mental models, and decision-making activities.

(this page intentionally blank)

A PREDICTIVE MODEL OF NUCLEAR POWER PLANT CREW DECISION-MAKING AND PERFORMANCE IN A DYNAMIC SIMULATION ENVIRONMENT

By

Kevin Anthony Coyne

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:
Professor Ali Mosleh, Committee Chair, and Advisor
Professor Thomas S. Wallsten, Dean's Representative
Associate Professor Michel Cukier
Associate Professor Jeffrey Herrmann
Assistant Professor Byeng Dong Youn
Dr. Nathan Siu, Special Member

Dedication

This work would have been impossible without the support of my loving, supportive, and – most importantly – patient wife, Beth Coyne. Equal admiration is due to my three children – Albert, Stephen, and Ben – for their support and willingness tolerate a father endlessly pecking at a computer keyboard. I also owe an enormous debt of gratitude to the work ethic and motivation inspired by my parents – Joseph and Louise Coyne – who demonstrated unwavering dedication to their children and showed the bounty that comes from focusing on the truly important things in life.

This dedication would not be complete without noting the vast contribution to my professional success that came from the mentoring, friendship, and advice from Jeff Manning of the Nuclear Engineering and Planning Department at Norfolk Naval Shipyard. Although I certainly do not miss working an endless parade of back shifts, Jeff's common sense approach to nuclear engineering has served as a solid foundation to all my professional endeavors. I would also like to recognize Dr. James L. Kelly, professor emeritus of nuclear engineering at the University of Virginia. Dr. Kelly not only provided me with first rate instruction in nuclear engineering during my undergraduate and graduate studies at UVA, but also inspired me to embrace learning as a life long mission.

(this page intentionally blank)

Acknowledgements

Financial support for this work was provided by a U.S. Nuclear Regulatory Commission Graduate Fellowship. As with any large project, without the support from a significant number of people, this research project would not have been possible. I would like to specifically acknowledge the assistance, technical advice, critical review, and encouragement from the following people and organizations:

University of Maryland
Professor Ali Mosleh and Dr. James Chang

U.S. Nuclear Regulatory Commission
Charles Ader, Christiana Lui, Dr. Nathan Siu, Dr. Erasmia Lois
Bruce Bartlett, Robert Krsek, Jay Lennartz,
James Shea and John Stang

Oak Ridge Institute for Science Education
Dr. Milton Constantin

Organization for Economic Cooperation and Development (OECD)
Halden Reactor Project
Andreas Bye, Helena Broberg, and Michael Hildebrandt

Paul Scherrer Institut
Dr. Vinh Dang and Davide Mercurio

Sandia National Laboratory
Dr. Ronald Boring

(this page intentionally blank)

Table of Contents

(this page intentionally blank)

List of Tables

(this page intentionally blank)

List of Figures

(this page intentionally blank)

List of Acronyms

ADAPT	Analysis of Dynamic Accident Progression Trees
ADS	Accident Dynamics Simulator
AFW	Auxiliary Feedwater System
AOO	Anticipated Operational Occurrence
CES	Cognitive Event Simulator
COSIMO	Cognitive Simulation Model
CRL	Cutoff Reinforcement Learning
DDET	Discrete Dynamic Event Tree
ECCS	Emergency Core Cooling System
EOC	Error of Commission
EOO	Error of Omission
EOP	Emergency Operating Procedures
ES	Event Sequence
F&B	Feed and Bleed
FRG	Functional Recovery Guideline
FW	FeedWater
HAMMLAB	Halden Human-Machine Laboratory
HRA	Human Reliability Analysis
IDAC	Information, Decision, and Action within a Crew context
LCO	Limiting Condition for Operation
LOCA	Loss of (Reactor) Coolant Accident
LOFW	Loss of Feedwater
MCDET	Monte Carlo – Dynamic Event Tree
MDAFP	Motor Driven Auxiliary Feedwater Pump
MSIV	Main Steam Isolation Valve
MSLB	Main Steam Line Break
NRC	Nuclear Regulatory Commission
PIF	Performance Influencing Factor
PORV	Power Operated Relief Valve
PRA	Probabilistic Risk Assessment
PZR	Pressurizer
RCS	Reactor Coolant System
RO	Reactor Operator
RPD	Recognition Primed Decision-Making
SG	Steam Generator
SGTR	Steam Generator Tube Rupture
SRO	Senior Reactor Operator
TDAFP	Turbine Driven Auxiliary Feedwater Pump
TMI	Three Mile Island Nuclear Power Station

(this page intentionally blank)

1. Introduction

1.1 Motivation

The safe operation of complex systems requires close coordination between the human operator and the physical hardware. For example, nuclear plant control room operators must efficiently perform complex tasks following an accident in order to maintain an adequate level of public safety. Oftentimes, operators face dynamic conditions with incomplete information and little time to consider options. Safety can be greatly improved if the conditions that could lead operators to make poor decisions and commit erroneous actions can be predicted and mitigated. Current techniques for predicting human errors largely rely on a static analysis of the tasks an operator must perform [1]. Unfortunately, static analyses cannot capture the dynamic factors and feedback loops that influence human behavior. A simulation-based approach that dynamically couples human and hardware performance may provide a better prediction of operator behavior. The primary goal of this research project is the development and validation of a nuclear power plant operator cognitive model. The secondary goal is the application of the cognitive model within a dynamic simulation environment in order to identify situational factors that can lead to human errors.

1.1.1 Operator Errors During Nuclear Plant Operations

Nuclear power plants present several unique hazards to the health and safety of the public. Foremost, the nuclear reactor core contains a substantial inventory of radioactive fission products that, if released, pose a serious threat to public health and the environment. Even after a reactor core shutdown, fission product radioactive decay continues to produce a substantial amount of heat that must be removed to prevent core damage and a radiological release. Although nuclear plants have automatic systems to prevent fission product release and provide core cooling, the operators play a vital role in ensuring plant safety. Numerous studies have shown that human error can be a significant contributor to the overall risk of nuclear power plants [2-4].

The 1979 accident at Three Mile Island (TMI) Unit 2 nuclear plant highlights several important factors that can create an error forcing situation. The TMI-2 nuclear power plant automatically shut down on March 3, 1979, following an unanticipated failure of the main feedwater system. Although this was an abnormal event, it is not an unexpected event and safety systems existed to mitigate this type of accident. Emergency feed water pumps started automatically, but misaligned valves prevented proper flow from this backup safety system. Immediately following the plant shutdown, a relief valve on the reactor coolant system opened but failed to fully close, resulting in leakage of reactor coolant. Despite these failures, the reactor core was adequately cooled due to the actuation of the emergency core cooling system. However, the operators failed to realize that (1) valve misalignments in the emergency feedwater system prevented makeup cooling water from reaching the steam generators, and (2) the reactor coolant pressure relief failed to close, resulting in leakage of coolant from the reactor core. The operators, erroneously believing that that too much water was being injected into reactor coolant system (due to their failure to correctly interpret plant symptoms), reduced the flow from the emergency core cooling system. The reduction in emergency core cooling system flow was an operator error that resulted in significant reactor core damage. The event was eventually terminated when a new operating crew arrived for shift change and identified the open relief valve and re-initiated adequate core cooling flow. The TMI accident was caused, in part, by failure of the operators to adequately perceive relevant plant information, form a correct mental model of the situational context, and execute actions that could safely terminate the event. Although human errors have not resulted in a serious nuclear accident in the United States since 1979, safety significant human error events continue to occur [2, 5].

Engineered Safety Feature Bypass Events

Nuclear power plants are equipped with a variety of engineered safety features designed to mitigate the consequences of accidents. Engineered safety systems typically support critical functions such as core cooling, fission product containment, and support services such as electrical power. Although these systems are often designed to operate automatically, they can be bypassed by the operators to allow maintenance, testing, and plant startups and shutdowns. In general, bypassing a safety system is strictly controlled by plant procedures since it can render plant safety systems inoperable. In 1995, the Nuclear Regulatory Commission (NRC) conducted a review of operational events involving the bypass of engineered safety features [5]. This review identified nine inappropriate engineered safety features bypass events during the period of 1991 through 1995. In each of these events, control room operators intentionally took actions that rendered key safety systems inoperable, contrary to both their training and procedural guidance. It should be stressed that these actions were not associated with malicious intent – rather, these events often resulted from the inappropriate resolution of conflicting operational goals. Although none of these events resulted in significant safety consequences, the inappropriate bypassing or inhibition of safety systems can have dire outcomes, as evidenced by the accident at Three Mile Island.

Human Error Contribution to Operational Events

Human errors have contributed to the risk significance of many operational events at nuclear plants [2]. The Nuclear Regulatory Commission recently sponsored a review of forty-eight risk significant operating events that occurred during the period of 1992 through 2000. In general, the operating events involved losses of electrical power or failures of emergency core cooling systems. Of the forty-eight events, researchers determined that human error significantly influenced the risk of thirty-seven events. Where it was possible to obtain quantitative results, the average human error contribution to the event risk increase was 62%. Errors by the control room operators were present in 54% of events. More surprisingly, for the events where human error was a factor, researchers determined that, in each case, four or more human performance issues had contributed to the risk significance. The study noted that further work should be done to better understand the risk impacts and linkage factors associated with event sequences containing multiple human errors.

Results from the U.S. NRC Individual Plant Examination Program

In 1988, the NRC issued Generic Letter 88-20, "Individual Plant Examination for Severe Accident Vulnerabilities," which requested that utilities identify plant-specific vulnerabilities that could be fixed with low cost improvements. Severe accidents are generally defined as events involving a substantial amount of nuclear core damage and represent challenges beyond the normal licensing basis of the nuclear plant. One goal of the Individual Plant Examination (IPE) program was to identify human actions important to severe accident prevention and mitigation. The results of the IPE program were summarized in NRC NUREG-1560, "Individual Plant Examination Program: Perspective on Reactor Safety and Plant Performance" [6]. Although the results for individual plants varied by a considerable amount, it was generally found that a relatively few number of human failure events could contribute significantly to overall plant risk. For example, a failure to align an alternate water source to the emergency core cooling system during a loss of coolant accident can contribute up to 17% of the total core damage risk for a pressurized water reactor. The failure to manually depressurize a boiling water reactor to allow low pressure systems to provide cooling water to the reactor core could contribute up to 45% of the total core damage risk. It should be noted that the IPE program was principally focused on errors associated with delays or omissions in the execution of procedural actions or recovery actions for failed equipment. Consequently, the IPE program did not attempt to capture the full range of human error possibilities.

Discussion

As just illustrated, human error is a significant contributor to the overall public health risk from nuclear power plants. In particular, errors associated with the bypass or defeat of safety systems and the aggregate impact of multiple human errors represent a particular challenge. It should be noted that these errors have not arisen from malevolent intentions on the part of the operator. Indeed, the operators responsible for operational errors either have not recognized that they have deviated from standard operating practices or believe that their deviation was in the best interest of plant safety. Consistent with this observation, deliberate acts of sabotage by the operators historically have not been included in nuclear plant risk studies. Furthermore, the Nuclear Regulatory Commission's probabilistic risk assessment procedure guide notes that "it is assumed that all plant personnel act in a manner they believe to be in the best interest of the plant" [7]. Therefore, the scope of operator actions that should be considered in a nuclear plant risk assessment can be generally reduced to those actions that arise through a systematic decision-making process by a well-intentioned operator. The ability to model knowledge-based behaviors that may be activated under inappropriate circumstances can improve the accurate prediction and mitigation of human performance errors and improve the overall safety of nuclear plants.

1.1.2 Need for Better Human Error Analysis Tools

Human reliability assessment (HRA) models are used to predict operator performance and can be classified in terms of their basic modeling assumptions.

Many of the so-called first generation HRA models were based on either time-reliability curves or simple information processing models [8]. For example, the Technique for Human Error Rate Prediction (THERP) [9] assigns nominal human error rates based on activity characteristics derived from a task analysis. These nominal error rates can then be modified by performance shaping factors that account for items such as operator experience level and stress. Second generation HRA models, such as the Cognitive Reliability and Error Analysis Method (CREAM), are built upon a stronger cognitive foundation [8]. Rather than assigning error probabilities based on functional tasks, the CREAM model first decomposes operator tasks into distinct cognitive activities (e.g., communication, diagnosis, monitoring) and human functions (e.g., observation, interpretation, planning, and execution). Human error rates are then determined based on the cognitive activity, human functions, and relevant performance shaping factors. Both the first and second generation HRA methods generally depend on a priori knowledge of the tasks that will be performed by the operator during an accident sequence. Because the scope of the task analysis supporting these methods is usually established by plant procedures, evaluating human error rates for actions outside the scope of the procedures presents a challenge.

Previous research efforts have found that nuclear plant operators can be induced to commit unsafe actions under certain error forcing situational contexts [10]. Situational context includes factors such as the system state, the operator's state of mind, and the sequence and timing of events. In particular, four important observations regarding accident analysis have been noted [11]:

1. *Plant operators and plant components are interacting parts of an overall system that responds to upset conditions.*
2. *The actions of operators are governed by their beliefs as to the current state of the plant.*
3. *The operators have memory; their beliefs at any given point in time are influenced by the past sequence of events and by earlier trains of thought.*
4. *A number of operators are involved during the accident.*

These observations point to the need to develop analysis techniques that can explicitly model the dynamics and feedback of nuclear systems while capturing the cognitive behavior and limitations of operators performing within a crew environment.

Although the dynamic interaction and feedback between man and machine strongly affects the situational context and potential for operator error, these dynamic effects are difficult to model with first or second generation HRA methods. Simulation-based HRA methods can address many of the shortcomings of earlier HRA methods. For example, the dynamic interaction and feedback resulting from operator actions can be directly modeled. The time-dependent behavior of performance influencing factors such as stress, fatigue, and work load can also be modeled within a simulation environment. Simulation-based methods can augment the data usually derived from time consuming and expensive control room simulator experiments conducted with actual control room crews. A computer simulation model can also explore a wider range of accident conditions than would be possible with actual control room operators in a simulator. An added benefit of simulation approaches is that a comparison between the simulation results and actual human performance on a similar task allows the analyst to benefit from feedback and develop greater accuracy and coverage [12]. Finally, a simulation approach allows a better determination of the consequences of a human error event. By coupling an operator model with a plant system model, one can determine the impact of each error event on the system – thus it becomes a straightforward matter to determine if a human error event has an actual risk impact.

The Accident Dynamics Simulator with the Information, Decision, and Action in a Crew context cognitive model (ADS-IDAC) provides a means to achieve these human error analysis goals. The main objective of this research effort is to improve the modeling and simulation capabilities of the ADS-IDAC code to support better simulations of human behavior.

1.2 Project Objectives

The main objectives of this research are the following: (1) development of a cognitive engine capable of representing the decision-making behavior of a nuclear plant operator; (2) implementation of the cognitive engine within the ADS-IDAC simulation environment; (3) validation of the ADS-IDAC operator, and (4) development of general ADS-IDAC analysis procedures for human performance prediction. These objectives and the associated research activities are described in the remainder of this section

1.2.1 Development of Operator Decision-Making Engine

The ADS-IDAC simulation model requires a cognitive engine to guide operator decision-making. The purpose of the cognitive engine is to form a situational assessment from perceived information (diagnosis), to identify and select suitable goals based on the situational assessment, to identify and select suitable strategies for obtaining goals, and to prioritize and resolve conflicts among the selected goal/strategy sets. The cognitive engine is based largely on a recognition primed naturalistic decision making (RPD) model [13, 14]. The RPD model attempts to simulate the behavior of experienced decision makers under time pressure. Because the RPD model is unable to capture all reasonable and expected operator behaviors; it was necessary to augment the RPD framework in order to capture the inherent variability in the human decision making process. For example, variations in operator response times and the selection of control values for key plant parameters were simulated with stochastic models. The following components were developed within the operator cognitive engine:

Information Filtering

A key feature of the ADS-IDAC simulation model is that all operator behaviors arise from perceived data rather than the direct output from the plant thermal-hydraulic plant model. Before an operator can use any plant information, the data must first pass through the operator's perception filter. Because the perception filter can either screen out or distort data obtained from the plant model, the operator may possess incomplete or inaccurate information. If the operator uses this incomplete or inaccurate data to guide his/her decisions and actions, human error events may occur. A goal of the ADS-IDAC project is to identify situations and contexts where operators may implement inappropriate knowledge-based actions due to limitations of the perception filtering process. Therefore, it is necessary to develop information filters capable of masking or distorting information under certain situational contexts.

Diagnosis Module

Another key feature of the cognitive model is the formulation of a situational assessment through a diagnosis process. Rasmussen has identified two basic diagnosis strategies – symptomatic search and topographic search [15]. A symptomatic search uses basic feature matching to identify memorized events that match the observed symptoms. The topographic search process seeks to identify differences between the actual system condition and the operator's mental representation of normal or planned operation. The symptomatic search strategy generally has a lower cognitive burden than the topographical strategy but is dependent on the ability to pattern match to a previously experienced event. Conversely, the topographic strategy has a higher cognitive burden, but is better able to deal with novel situations. A diagnosis module that uses a fuzzy logic approach to perform a symptomatic feature matching process has been developed [16]. To support a topographic search strategy, it is necessary to develop a reasonable mental representation of the nuclear plant system within ADS-IDAC. One approach of creating a mental representation of the nuclear plant, suggested by Lind, is the use of mass and energy conservation laws [17, 18]. Consequently, a major focus of this activity is decomposing the plant into energy and mass flows that are consistent with typical operator mental representations of the reactor plant.

Rule Sets, Goals, Strategies, and Actions

A major component of this research process was the identification of heuristic rules used by operators to direct behavior. Under the RPD paradigm, activation of a particular rule or rule set will be determined by the operator's situational assessment. Therefore, in addition to rule set identification, it is necessary to catalog the situational context associated with each rule. For example, a possible heuristic rule might be "prevent overfill of the reactor coolant system by reducing makeup flow." Under normal operating conditions, application of this rule could prevent an unnecessary loss of pressure control and a challenge to plant safety relief valves. However, during the Three Mile Island accident, application of this rule led directly to a core melt event. A lesson learned from the Cognitive Event Simulator (CES) research (see Section 2.4.3.1) was that while an extremely detailed rule set might provide good performance under certain conditions, it does not adapt well to new situations and may not provide a reasonable representation of human behavior. Consequently, an objective of this activity is the development of a minimum rule set that can reasonably represent operator behavior.

Memory Management and Goal Prioritization

ADS-IDAC includes a memory model that includes working memory, intermediate memory, and a knowledge base. In order to effectively implement a memory model, it is necessary to develop better methods to group, prioritize, and utilize memorized information. One possible approach is to group information based on a simple surface similarity approach (i.e., information is grouped by the associated system). However, expert decision makers tend to utilize structural similarities to group information [19]. Structural similarity refers to the underlying physical principles associated with the information. Thus, a goal of this activity is to develop a method of grouping information in a manner more consistent with an operator's mental representation of the reactor plant.

Performance Influencing Factors

ADS-IDAC employs both static and dynamic performance influencing factors (PIFs) to influence and shape operator behavior. As the name suggests, static PIFs are constant parameters intended to represent the fixed environmental and organizational factors that affect crew behavior. Conversely, dynamic PIFs reflect transient conditions and model variations in the operator's mental state during a scenario. The IDAC model includes fifty performance influencing factors (PIFs) which can be used to influence operator behavior [20]. The IDAC model includes the following broad categories of PIFs:

Internal Influences

- Mental State – associated with the operator's cognitive and emotional state;
- Memorized Information – associated with perception and recall;
- Physical Factors – related to ergonomic conditions and the operator's physical abilities.

External Influences

- Team-Related Factors - pertains to coordination requirements among crew members;
- Organizational Factors – associated with management influences on behavior;
- Environmental Factors - relates to environmental conditions that affect behavior (harsh environment, physical access, etc.);
- Conditioning Events - unanticipated changes of system state.

Due to time, resource, and data limitations, this project does not attempt to implement and validate the effects of all fifty PIFs on the operator cognitive model. Instead, this project focused on implementing and validating a small subset of PIF factors that are necessary to capture a reasonable range of operator behaviors.

1.2.2 Calibration of the Operator Model

Even when personnel selection procedures, training programs, and administrative programs are consistently implemented, nuclear plant operators can exhibit significant crew-to-crew performance variabilities. These performance variabilities can arise from differences in crew knowledge, skills, and experience; crew specific organizational factors; and operator preferences and tendencies. Within a dynamic risk assessment, variability is modeled by introducing new branches into a dynamic event tree. The path along a set of serially connected branches defines a specific accident scenario. Branches can be generated to simulate slow or fast procedure execution speed, skipping of procedure steps, reliance on memorized information, activation of mental beliefs, variations in control inputs, and equipment failures. A calibration process must be developed to represent observed crew behaviors within a generalized ADS-IDAC branching rules and knowledge framework. The goal of this process is to characterize crew behavior in a manner that preserves generalizability without the development of overly prescriptive branching rules.

1.2.3 Validation Activities

In order to verify, to the extent possible, that the ADS-IDAC is capable of appropriately simulating operator behavior, the research project also involved a validation component. In this project, the validation process focused on validating the results and outcomes of the model rather than its functional details. Typical validation concerns include face validity, content validity, and criterion related validity [21]. Face validity refers to the degree that the model captures important and relevant behaviors as judged by potential users. By its nature, face validity is a largely subjective and weak measure of overall validity, but can influence the level of confidence that users have in model results. Content-related validity refers to the inclusion of all pertinent factors that can influence the capability of the model to meet its objectives. In this case, content-related validity implies that the ADS-IDAC model addresses factors that significantly influence the identification of nuclear plant error forcing situations. Criterion-related validity refers to the capability of the model to predict operator behaviors and, where possible, provide results consist with other modeling techniques.

1.2.4 Development of General Analysis Procedures

In order to support implementation of the ADS-IDAC approach for human reliability analysis, general procedures must be developed. These procedures are needed to provide a consistent approach to creating and validating the operating knowledge base, developing a realistic representation of the reactor plant system, and formulating inputs needed to implement the ADS-IDAC approach.

1.3 Overview of Dissertation

This dissertation report describes the background, development, implementation, calibration, and validation of the ADS-IDAC simulation approach.

Chapter 2 provides a general overview of the theoretical background supporting ADS-IDAC, and Chapter 3 provides a description of the ADS-IDAC model. Chapters 4, 5, and 6 describe the implementation of information, decision-making, and action modules of the IDAC model in the ADS-IDAC simulation code. Chapter 7 describes the implementation of the static and dynamic performance influencing factors that affect operator behavior. Chapter 8 describes branching rules used to construct a dynamic event tree. Chapter 9 describes the calibration activities used to ensure that ADS-IDAC realistically modeled operator behavior, and Chapter 10 describes the validation of the model. Chapter 11 summarizes the main conclusions of this project and provides recommendations for future work. A number of appendices are also included that document features of the ADS-IDAC operator knowledge base, provide detailed simulation results, and describe the operation of the simulation code.

1.3.1 Theoretical Foundations for Operator Modeling (Chapter 2)

This Chapter discusses the basic theory underlying human cognition and decision making, as it applies to the ADS-IDAC project. Specific topics discussed include the decision-making context for a nuclear plant control room; a summary of the literature pertaining to cognitive modeling and decision-making; a summary of current human reliability analysis approaches; and a summary of the contributions of this research project to the state of knowledge in human performance modeling.

1.3.2 ADS-IDAC Overview (Chapter 3)

Chapter 3 describes the main components of the ADS-IDAC simulation model. Included in this section is an overview of computer code architecture, code input requirements, and the output data generated by ADS-IDAC.

1.3.3 Information Processing Module (Chapter 4)

Chapter 4 describes the Information processing module, the first step in the Information, Decision-Making, and Action execution (IDA) cognitive model. Main topics include information gathering and processing, and the content of the operator knowledge base. Specific features of the operator knowledge base include a description of the plant functional decomposition – the method used to link plant components to specific plant safety functions; knowledge based actions and mental beliefs; and the modeling of plant operating procedures.

1.3.4 Decision Making Module (Chapter 5)

The main features of the ADS-IDAC decision-making engine are described in Chapter 5. This section includes a description of the goal and problem solving strategy selection processes; the use of discrete mental beliefs to model more complex forms of operator interactions with the nuclear plant model; and the diagnostic module that supports the operator's situational awareness.

1.3.5 Action Execution Module (Chapter 6)

The Action execution process, the last step of the IDA cognitive model, is described in Chapter 6. This section discusses the modeling of operator actions, including methods for handling crew variability; and the procedure step-skipping module. The ADS-IDAC procedure step-skipping, a type of an error of omission, illustrates how the information gathering processes, the operator knowledge base, and other factors that influence operator behavior can reflect the dynamic contextual factors that can lead to operator errors.

1.3.6 Performance Influencing Factors and Crew Model (Chapter 7)

Human performance can be influenced by many factors, including the external environment (e.g., lighting, temperature, and noise), ergonomics (e.g., control panel layout, clarity of procedures), mental factors (e.g., stress, workload, knowledge, experience), and organizational influences (e.g., assigned responsibilities, peer checking). Based on the time scale selected for the analysis, these factors can be considered to be either static or dynamic. Chapter 7 provides an overview of the performance influencing factors included within the ADS-IDAC model and the behaviors they influence.

1.3.7 Dynamic Event Tree Construction (Chapter 8)

Chapter 8 provides an overview of dynamic event tree construction during an ADS-IDAC simulation. Guiding dynamic event tree branching through the use of branching rules is the main technique for simulating variabilities in crew behavior. The main categories of branching rules are described, as are methods used to focus computational effort on more significant aspects of the analysis.

1.3.8 Implementation of ADS-IDAC Approach (Chapter 9)

Chapter 9 describes the general procedures developed to realistically model nuclear plant thermal hydraulic systems, calibrate the IDAC human performance model, and apply ADS-IDAC to the prediction of sources of crew variability. An underlying assumption of the ADS-IDAC approach is that deviations from the normative (or expected) set of operator behaviors following an accident can sometimes lead to a degraded plant state. This section also discusses methods developed to process the results of experiments using actual plant operators into a form suitable for incorporation into the ADS-IDAC knowledge base.

1.3.9 Validation and Verification (Chapter 10)

Chapter 10 describes the validation efforts used to demonstrate the utility of ADS-IDAC to the analysis of human performance issues during nuclear power plant accidents. Three forms of validation are discussed: a comparison of the implemented

ADS-IDAC model to other accepted HRA techniques (content validity), demonstration that ADS-IDAC generates realistic and reasonable results for a variety of accident scenarios (face validity) and an illustration of the predictive capabilities of ADS-IDAC using experimental data from the Halden Reactor Project simulator (criterion validity).

1.3.10 Conclusions (Chapter 11)

Chapter 11 summarizes the key conclusions arising from this research effort and identifies several recommendations for future work.

2. Theoretical Foundations for Operator Modeling

This section begins with an overview of the ADS-IDAC model prior to the start of this research project. This information provides the starting point for this research effort. Because environmental and organizational factors can influence human behavior, a description of the nuclear plant control room environment is then provided. This is followed by a discussion of human error taxonomies and their relationship the underlying cognitive model used to define human error. In order to place the contributions of this research into proper perspective, a general overview of human reliability analysis methods is provided, with particular emphasis on the use of simulation approaches to human error analysis and prediction. Finally, the specific contributions arising form this research effort are described.

2.1 Overview of the ADS-IDAC Model

The IDAC (Information, Decision, and Action in a Crew context) cognitive model serves as underlying framework for operator behavior. IDAC decomposes the operator’s cognitive flow into three main process: information processing, decision-making, and action execution [22]. The domain of applicability of IDAC is constrained to environments characterized by high levels of training and explicit requirements to follow procedures [23]. These constraints simplify the modeling by limiting the degrees of freedom from the broader human response spectrum. In IDAC, the crew is modeled as a team of individuals working on different assigned tasks and communicating with one another. The individuals differ by the content of their memory, by their mental state, and by the goals and strategies they employ.

The IDAC model includes the following cognitive processes (see Figure 1):

- Information processing includes the capability of filtering incoming information to simulate the limitations of human perception. An operator must actively attend to incoming information in order for the filtering and perception process to be initiated.
- In the decision-making process, the operator develops a situational assessment of the current plant state based on perceived information. Because an operator may not reach a conclusion about the plant state with complete certainty, the cognitive model can assign a "degree of belief" (or subjective probability) about the operator's confidence that the perceived plant state is correct. Based on the perceived plant state, the operator will generate high level goals to guide further activities. The operator attempts to achieve the goals by implementing an appropriate problem-solving strategy.
- In the action execution process, the operator selects actions consistent with the goal implementation strategy.

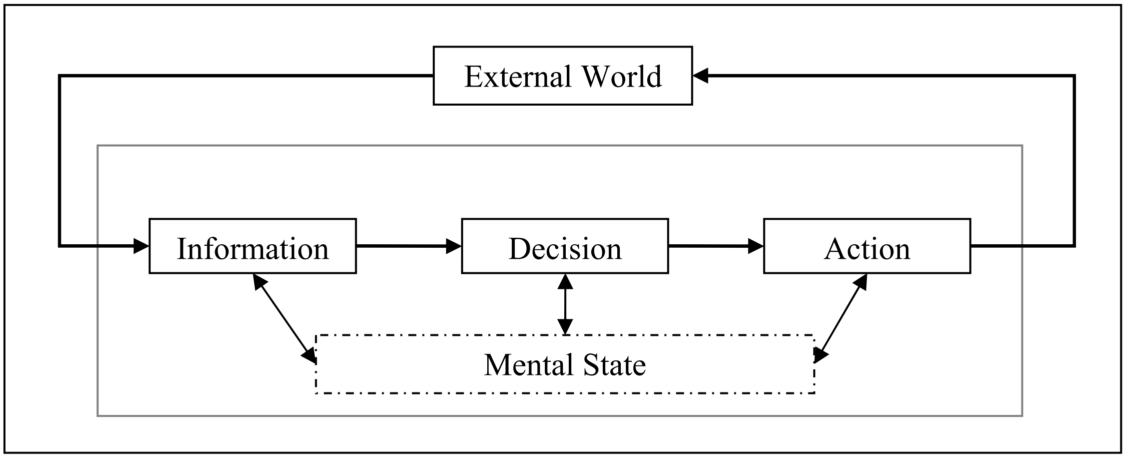


Figure 1- Overview of the IDAC Cognitive Model

These information, decision, and action processes are supported and influenced by the operator's mental state and memory. The operator's mental state includes factors such as cognitive mode, emotional state, and physiological stressors. Memory is decomposed into three main areas: working memory, intermediate memory, and the knowledge base. Working memory has a small information storage capacity and represents the information the operator is currently processing. Intermediate memory has a larger information storage capacity than working memory and is used to store information that might later be needed in working memory. The knowledge base contains the factual information about plant performance, procedures, diagnosis, and engineering principles that an operator is expected to know.

In 2004 Mosleh and Chang [24] noted the following specific research needs for ADS-IDAC:

- a stronger basis for cognition rules associated with information processing, and for selection of goals and problem solving processes;
- a more explicit representation of operator memory;
- improvements in the linkage between plant dynamics and the operator's mental state;
- development of an operator knowledge base that includes general rules of behavior for responding to accidents, a representation of the functional and physical characteristics of plant systems and related hardware; and
- validation of the operator behavior model.

This research project attempts to address each of these issues in order to improve the overall capabilities of ADS-IDAC for the prediction of human behavior.

2.2 The Nuclear Control Room Environment

Nuclear plant control rooms are designed to provide a large amount of information to the operators. Much of this information is associated with power production and economic considerations (e.g., power output, plant efficiency), but a significant portion is intended to provide safety-related information to the operators. Audible or visual alarms are usually provided for key parameters, but some control rooms have well over one thousand separate alarms. Thus, information overload during an abnormal event can become a problem. To reduce the potential for overload, operators attempt to focus on the most important parameters based on their perceived situational assessment. However, focusing on information believed to be important can have the undesired effect of filtering out relevant information if the operator's situational assessment or mental model of the plant is incorrect or incomplete.

To ensure public safety, the activities of licensed operators are generally guided by the license conditions specified for the facility, technical specifications, and written procedures. License conditions include technical and administrative requirements that must be met by the facility in order to comply with regulatory requirements. Technical specifications specify safety limits and limiting conditions for operation (LCOs). LCOs specify functional requirements for instrumentation, key process variables, and safety-related equipment. Quality assurance requirements for nuclear power plants (10 CFR 50, Appendix B) require that activities affecting the capability of plant equipment to perform their intended functions be performed in accordance with documented instructions or procedures. Written procedures have been developed to cover most anticipated normal, abnormal, and emergency conditions. However, it is still possible for situations to arise that are not adequately covered by procedures. Therefore, operators are expected to continuously evaluate the efficacy of procedures in use to determine if the plant state is improving or degrading. If a procedure fails to effectively mitigate an accident, it may be necessary for an operator to deviate from the procedure or change to a different procedure. Indeed, NRC regulations permit a licensed operator to deviate from license conditions, technical specification requirements, and procedures when necessary to protect public health and safety, provided that a licensed senior reactor operator has approved the deviation. Consistent with this approach, simulator studies of control room operators have found that operators use a number of higher level cognitive processes such as situational assessment and response planning even when following an emergency procedure [25].

Control room decisions often must be made under significant time constraints. Although many emergency safety systems are designed to automatically actuate, it is often necessary to realign safety systems to alternate or backup water or power supplies shortly after an accident. Additionally, operators must complete some critical emergency actions within a relatively short time period. High workload during accident conditions can further influence an operator's ability to perform actions promptly and accurately. Finally, all nuclear plant operators must work within a crew environment. Within the crew structure, the individual operators may not have a holistic view of the plant situational context. While the senior operators are expected to maintain a holistic view of the plant context, the crew members must communicate effectively to obtain this same level of understanding.

2.2.1 Operator Description

In the U.S., nuclear power plant operators are licensed by the Nuclear Regulatory Commission and regulated under the provisions of Title 10, Part 55, "Operators' Licenses," of the Code of Federal Regulations (10 CFR 55). The regulations recognize two qualifications for a licensed operator: operator and senior operator. Only a licensed operator or senior operator is permitted to manipulate the controls of a nuclear facility. Additionally, only a licensed senior operator is permitted to direct the activities of licensed operators. Regulations also establish minimum staffing requirements for a nuclear plant control room.

In order to obtain a license, an operator must meet certain medical requirements and pass written and operating tests to determine if he or she can operate the plant competently and safely. Operators generally receive training using realistic control room simulators with sufficient fidelity to ensure that simulated control manipulations can be performed using the same procedures used in the actual plant. Following initial licensing, operators must periodically stand control room watches and must pass a requalification program every two years. Additionally, operators are subject to fitness for duty requirements (e.g., abstinence from alcohol prior to assuming the watch and drug screening).

On the basis of their high level of experience and training, most nuclear plant operators could be categorized as experts in their field. A number of research studies have identified important differences in the decision-making and problem-solving behavior of experts versus novices. These differences include the following [19]:

- Experts tend to store knowledge in large interconnected units while novices tend to store information in small units. Consequently, experts are able to quickly access information needed to solve problems.
- Experts develop mental representation of a problem based on underlying structural principles while novices tend to represent problems based on the surface appearance of the problem.

- In the area of strategic knowledge, experts tend to work problems from the "givens" to the unknown, while novices tend to work from the unknowns back to the "givens."

Each of these factors has implications for how a nuclear operator develops a mental representation of the reactor plant and approaches the decision-making process. For example, these factors tend to support an ADS-IDAC knowledge base that links related information using the underlying structural principles of the nuclear plant complex. Additionally, an appropriate diagnostic process model for a nuclear plant operator would tend to start with perceived information and lead to a possible event diagnosis; rather than working backwards from a possible event diagnosis and verifying the existence of supporting information.

2.2.2 The Control Room as a Decision Production Environment

The IDAC model decomposes the operator's problem solving and decision-making into three main processes: information pre-processing (I); decision-making (D); and action execution (A) [22]. The information pre-process includes information filtering, comprehension and retrieval, relating and grouping, and prioritization. The decision-making process includes situation assessment, diagnosis, and response planning. The action execution process implements the actions identified from the decision-making process. In working through these cognitive processes, the operator should consider the following:

- Information Processing and Situational Assessment – An operator develops a situational assessment of the plant state by perceiving and interpreting information from the plant. However, plant information can be filtered by sensing and mental processes, such that perceived information is not identical to the actual state of nature. Additionally, instruments can fail or provide misleading information under certain conditions. Perception, filtering, interpretation, and instrument failures can all result in information conflicts. An operator might be biased to resolve these conflicts by discounting information that does fit into the current situational assessment. The end result of this process is a determination if the plant state is abnormal and, if so, a diagnosis of the problem.

- Selection of Appropriate Goals and Objectives – A high level goal is used to develop strategies and actions to mitigate an abnormal event. Examples of goals might include "maintain the plant at power", "shutdown the reactor", "shutdown, cool down, and depressurize the reactor plant", or "repair failed instrument." An operator must formulate an accurate situational assessment of the current plant state in order to identify appropriate goals and objectives. If the operator develops an incorrect situational assessment due to information filtering, misperception, or misinterpretation, it becomes more likely that inappropriate goals or objectives will be selected. Since goals and objectives will drive the selection of follow up actions, the selection of an inappropriate goal could result in an error event. The importance of goal selection is echoed by research studies of aircraft accidents that have concluded that goal selection errors were found to be the most frequent cognitive error in fatal accidents [26, 27].
- Selection of an Appropriate Mitigation Strategy – Once an operator selects an appropriate set of goals; strategies are selected to achieve these goals. Typical

strategies might include “follow procedure”, “actively gather information”, “problem solve using inductive or deductive reasoning”, “ask for advice,” or “wait and monitor.” If an operator has a high degree of confidence in the situational assessment and has determined that existing procedures can mitigate the condition, a “follow procedure” strategy would be a likely selection. However, if the operator lacks sufficient information to adequately assess the situation, he or she might opt for either a “wait and monitor” or “actively gather information” strategy to obtain additional information about the plant state before taking further action.

- Resolution of Conflicting Goals or Strategies – The desire to maintain a nuclear plant in operation to maintain critical power infrastructure or maximize economic gain must be balanced with a desire to maintain public health and safety. A decision to shutdown a plant carries a high economic cost for the utility, particularly if a shutdown was not needed. However, a decision to continue to operate when a plant shutdown is required can carry even higher economic and social costs. Goal conflicts can also arise in more mundane decisions, such as deciding whether or not to start a complicated test a few hours before a scheduled shift change. Waiting until after a shift change might cause significant schedule delays and increase economic costs, but starting the test before the shift change could increase the time pressure on the operators and increase the likelihood of errors.
- Prioritization and Sequencing of Goals or Strategies – The decision-making process for an event may generate multiple goals, each with two or more

associated actions. For example, during a complete loss of electrical power event, an operator might generate goals to provide adequate reactor core cooling, restore a source of electrical power, and prevent damage to plant equipment. Although there might not be any conflicts associated with these goals, workload limitations might require an operator to prioritize which goals will receive the greatest degree of attention. There might also be conditions where the advancement towards the highest priority goal is temporarily blocked – in this case, the operator could shift resources and attention to a lower priority goal that is achievable. Operators must prioritize goals and actions in a rational and adaptable way such that highest priority goals receive sufficient attention but efficient progress can be made toward the achievement of all goals.

- Selection of Appropriate Actions – there are often multiple methods of accomplishing certain actions. An action can be selected on the basis of its relative costs and benefits, complexity, prerequisites needed to be met to accomplish the actions, or personnel preference. For example, emergency cooling water for the nuclear plant can often be obtained from either a limited source of high purity water or an unlimited source of low quality water (e.g., sea water). High purity water can be used without causing damage to plant components, but may not be sufficient to meet safety needs. The unlimited low quality water supply can meet all immediate safety needs, but may result in excessive corrosion and long term damage to plant components. The operator must balance short term safety goals against longer term economic goals when selecting an appropriate course of action.

In any decision-making environment, several factors can impact the decision-making process. These may include the following [28]:

- Values which define factors that are important to the decision maker;
- Objectives which define the decision-maker's desired goal states;
- The certainty (or uncertainty) of information that must be considered; and,
- The possible consequences of the decision.

The decision-making process can consist of either a single decision point or a sequence of decisions. If the process is a sequence of decisions, decisions early in the process can obviously affect decisions that must be made later in the process.
In the case of a control room decision-making, a series of decisions must be made with uncertain information. Because of the significant public health impact that a nuclear accident represents, the consequences from each decision could be significant. Factors such as values and objectives will largely depend on the personnel preferences of individual operators. However, the personnel selection and training programs for nuclear operators should instill some level of consistency in values and objectives.

Although the control room decision-making environment is somewhat structured due to the regulatory requirements imposed on a nuclear plant, many decisions must still be made based on the judgment and experience of the operators. In particular, accident scenarios that have not been experienced in the past or have not been a focus of training activities may place operators in an unfamiliar decision-making mode. These ‘unstructured’ decisions might place operators at a higher error potential. Mintzberg, Raisinghani and Théoret have examined the decision-making process associated with so-called ‘unstructured’ decisions [29]. Briefly, unstructured decisions are decisions that have not been encountered in quite the same form and for which no predetermined or ordered set of responses exists in the organization. A framework for unstructured decision making was developed that identified three phases of decision-making: identification, development, and selection (see Figure 2). Identification is primarily concerned with the decision-maker recognizing that a problem exists. Development consists of a diagnosis routine followed by activities to identify possible solutions. Candidate solutions can arise from either a search and screening process to identify solutions that have been used for other similar problems or a problem-specific design process to develop a unique solution for the current problem. The selection process is characterized by an evaluation-choice routine followed by an authorization process to approve the selected alternative. Evaluation can be performed by any of three of possible processes: judgment (where a single decision-maker simply makes up his/her mind based on experience), bargaining (where a decision is reached by a group of decision makers with competing goals), and analysis (based on a factual evaluation of candidate alternatives).

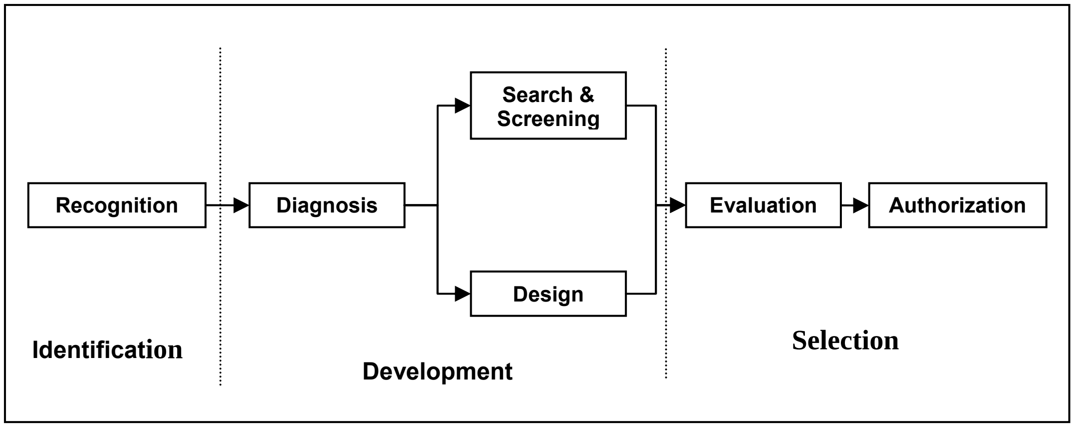


Figure 2 - Simplified Unstructured Decision Production System

Although this framework was based on case studies of long-term decision making processes (lasting from one to four years), this model is still useful for identifying some key elements in control room decision making. In particular, the recognition phase corresponds to the information block of the IDAC cognitive model. The development and selection phases correspond to the decision block of the IDAC model. Because the time frame of interest for the ADS-IDAC simulation is the first few hours of an accident, it is not expected that operators would attempt to identify a candidate solution through design; rather they will focus on a search and screening of existing solutions.

After the first few hours of an accident, the control room decision making structure will change dramatically as emergency response facilities are activated. In general, command and control of the accident response will shift from the control room to an offsite emergency operations facility. The activation of additional onsite emergency response facilities such as a technical support center and an operations support center will further change the control decision making dynamics. For this reason the ADS-IDAC simulation model should only be considered to be valid when the primary decision making responsibility resides in the control room.

2.3 Human Error Overview

In order to predict situations that may lead to human error events, it is first necessary to define specifically what is meant by the term “human error.” A clear definition is needed to ensure that human error data, when available, can be appropriately used, and that the results from a human reliability assessment can be unambiguously communicated. Proposed definitions include:

> *…all those occasions in which a planned sequence of mental or physical activities fail to achieve its intended outcome, and where these failures cannot be attributed to the intervention of some chance agency [30];*

> *… unwanted actions or inactions that arise from problems in sequencing, timing, knowledge, interfaces, and/or procedures that result from deviations from expected standards or norms that places people, equipment, and systems at risk [31].*

Other researchers have resisted providing a definition of human error. For example, Rasmussen views to human error as man-machine or man-task misfits [32], but also notes that human error is not a stable category of events [33]. Hollnagel also resists defining “human error” and instead stresses the need to link the operator model and the error classification scheme in order to better separate actions, causes, and consequences [8]. Thus, although a specific definition of error is somewhat elusive, there are some common elements among the various treatments of human error events. Notably, human error events arise from an underlying cognitive process and produce undesirable consequences. To understand the causes of human error, it is necessary to link the actions that lead to an error event back to the underlying cognitive model. In order to better define the term “human error,” the following sections provide an overview of commonly used human error taxonomies. Error taxonomies provide a hierarchical structure for classifying error events and collecting error rate data.

2.3.1 Errors of Commission and Omission

The concept of errors of commission and omission can be seen as an artifact of modeling human reliability within a probabilistic event tree framework [34]. Within this logical structure, human behavior is often modeled as the response to two distinct questions: (1) did the operator perform an action, and (2) was the correct action performed? A failure to act is generally called an error of omission, while performance of an incorrect act is termed an error of commission. NUREG-1792, “Good Practices for Implementing Human Reliability Analysis (HRA),” provides a more formalized definition of these error types [35]:

Error of Omission (EOO): A human failure event resulting from a failure to take a required action, that leads to an unchanged or inappropriately changed plant configuration with the consequence of a degraded plant state.

Error of Commission (EOC): A human failure event resulting from an overt, unsafe action, that, when taken, leads to a change in plant configuration with the consequence of a degraded plant state.

Although the potential for EOCs has long been recognized, explicit modeling of errors of commission has generally been beyond the scope of commonly used probabilistic risk assessment practices [35]. One difficulty in modeling EOCs is identification and evaluation of the potentially infinite number of actions that an operator might execute in response to an accident. Conversely, EOOs are somewhat easier to model since they are generally constrained by the scope of the facility operating procedures. The EOO/EOC taxonomy largely distinguishes between the types of human behaviors that have been historically included in risk assessments rather than linking errors to an underlying cause. Although a goal of cognitive modeling approaches such as ADS-IDAC is to provide a means to identify potentially risk significant EOCs, the EOO/EOC taxonomy lacks sufficient detail to adequately communicate the results of a cognitive human error model. In fact, researchers such as Hollnagel argue that the error of commission and omission taxonomy fails to distinguish clearly between error cause and manifestation [34]. Thus, it is often more useful to define human error in terms of the underlying cognitive processes rather than simply the observed consequence.

2.3.2 Lapses, Slips and Mistakes

The lapse, slip and mistake error taxonomy links human errors to the desired goal of an action. Lapses are generally associated with forgetting information relevant to the formulation of a goal or the execution of an action. Slips represent a discrepancy between the intended goal and overt action while mistakes are associated with the selection of an inappropriate goal [36]. Norman identifies six categories of slip type errors:

- Capture Errors – caused when two or more action sequences share common starting elements. The more frequently used sequence can "capture" or override a less frequent, but intended, action sequence;
- Description Errors – caused when the mental description of an action is ambiguously defined, leading the operator to act upon an unintended object that fits the ambiguous description;
- Data-Driven Errors – caused when nuisance data overrides correct data;
- Associative Action Errors – caused when an external trigger, common to two or more action sequences, causes the activation of an inappropriate action sequence;
- Loss-of-Activation Errors – caused when the activation of the goal associated with an action sequence has decayed, resulting in termination of the action sequence;

- Mode Errors – typically associated with the operation of equipment that has two or more operating modes. An error occurs when the selected operating mode is not appropriate to the current situation.

Because slip errors manifest themselves as a discrepancy between the desired goal state and the result of human action, detection of a slip is normally possible provided that suitable feedback is provided to the operator. Conversely, the actions taken as a result of a mistake will be consistent with the selected (though inappropriate) goal. Thus, detection of a mistake can be difficult, even if feedback is provided to the operator. Although the lapse/slip/mistake taxonomy is relatively simple, it has stronger cognitive foundation than the EOO/EOC taxonomy. It is worth noting that both EOOs and EOCs can result from either a slip or a mistake.

2.3.3 Skill, Rule, and Knowledge Based Errors

Rasmussen views human errors as the result of man-machine or man-task misfits [32]. Systematic misfits are typically considered design errors, while occasional misfits due to the variability of the human operator are considered human errors. Rasmussen constructed an error taxonomy based on a cognitive model that includes the active intentions, expectations, and subjective goals of the operator. Rasmussen's taxonomy identifies three hierarchies of behavior:

- Skill-Based: governs the largely subconscious performance in controlled situations associated with stored patterns of behavior. Errors arise from variability in action execution or coordination issues.

- Rule-Based: governs performance in familiar situations controlled by stored rules of behavior. Because rule-based behavior is used to control skill-based performance routines, the error mechanisms related to skill based behavior remain active in this domain.
- Knowledge-Based: governs performance in unique, unfamiliar situations for which actions must be planned from an analysis and the prioritization of various goals. Because knowledge-based behavior is used to activate stored rules, the error mechanisms associated with both rule- and skill- based behavior remain active in this domain.

The skill-, rule-, knowledge-based model also includes performance shaping factors such as social climate, physiological stressors, physical workload, and emotional factors that can condition the operator's response. In order to classify an error event within this framework, Rasmussen developed a series of questions to determine if an error event was skill, rule, or knowledge driven:

1. Is this a routine situation for which the operator employs a highly skill-based procedure? Errors could arise from manual action variability, topographic disorientation (the act is performed on the wrong object), or stereotopic take-over (another highly skilled act interferes with the activity – similar to Norman's capture errors).
2. If the situation is not routine, does the operator recognize this fact? Errors might arise if the operator fails to identify the need for action (stereotopic fixation).

3. Is the situation covered by normal work practices or planned procedures? Errors might arise if the operator fails to recognize that an existing rule covers the situation, fails to recall an appropriate procedure accurately, or selects an inappropriate procedure.
4. If the situation is not routine or covered by existing rule-based procedures, does the operator recognize that knowledge-based response is needed? Errors might arise if the operator attempts to apply a familiar rule-based action in an inappropriate situation.
5. When a knowledge-based response is needed, does the operator collect information of sufficient quality and quantity? Errors might arise if the operator attempts to formulate a knowledge-based response with incomplete or misinterpreted information.
6. For a knowledge-based response, does the operator correctly perform functional analysis and deduction? Errors might arise the operator cannot adequately formulate a response.

Reason later combined the error taxonomies developed by Norman, Rasmussen, and others to create the Generic Error Modeling (GEM) system [30]. In the GEM system, Reason links slips, lapses, and mistakes with Rasmussen's skill-, rule-, and knowledge-based framework. This resulted in three basic error types: (1) skill-based slips and lapses, (2) rule-based mistakes, and (3) knowledge-based mistakes. The GEM system also explicitly considers the relative timing of an error event. Errors that occur before the detection of a problem are generally termed monitoring failures involving slips and lapses. These skill-based slips and lapses are usually associated with the operator either being inattentive or over-attentive to the process being monitored. Errors that occur after problem detection are generally called problem solving failures and involve rule- or knowledge-based mistakes. Reason states that a defining condition for a rule- or knowledge-based mistake is the operator's recognition that a problem exists. Rule-based mistakes are typically due to the misapplication of good rules (i.e., rules that have worked in the past but are applied to an inappropriate situation) or the application of bad rules. Knowledge-based mistakes are typically caused by bounded rationality (i.e., the inherent limitations of the human cognitive and memory processes) or the presence of incomplete or inaccurate information.

A key feature of the GEM system is the preference given to rule-based problem solving based on pattern recognition. Behavior rules are arranged in a hierarchy from general to specific. When an exception to a general rule is encountered, the operator will formulate a more specific rule. Because general rules tend to be applicable to a wider range of situations, they often have stronger activation then more specific rules. The model proceeds to a knowledge-based mode only when successive attempts to find a rule-based solution fail.

2.3.3 Errors in the IDAC Modeling Framework

Error within the IDAC framework is defined in terms of the human operator failing to meet the needs of the nuclear plant system [23]. The error taxonomy for the IDAC model is based on mismatches between internal and external reference points. Internal reference points refer to cognitive processes within the IDAC model and include information collection, diagnosis, decision, and action processes. External reference points are defined as the plant system, procedures, and the operator. Using these reference points, it is possible to identify five broad human error categories [37]:

- Plant-Crew Mismatch: erroneous or incomplete information from plant or operator observation error;
- Procedure-Plant Mismatch: erroneous or incomplete procedure;
- Crew-Plant Mismatch: diagnosis, decision, or execution errors;
- Crew-Procedures Mismatch: procedure inadequacy from a human factors viewpoint or crew lacking knowledge to understand procedure; and,
- Crew-Crew Mismatch: erroneous or incomplete communication.

A more detailed error taxonomy can be created by decomposition of each of the three main cognitive processes in the IDAC model (i.e., information processing, decision-making, and action execution):

- Error in information collected due to receipt of incomplete information from the plant or from another crew member or information filtering error;
- Incorrect or incomplete assessment of situation or solution to problem due to failure to adequately define the problem or error in problem solving strategy selection;

- Decision error due to inappropriate selection of solution from equivalent alternatives or selection of incorrect decision criteria;
- Error in action execution due to high operator workload or poorly human factored environment.

A taxonomy classification and data collection scheme has been developed based on these broad error types [23]. A unique feature of the IDAC error taxonomy is the association of human error with the failure to meet a plant need. In some sense, this classification scheme serves a screening function in that human errors that do not result in the failure to meet a plant need would not be subject to further evaluation. This serves to focus error analysis on the actions that most impact the safety of the plant system. Although defining the "plant needs" can still present a challenge, use of this framework within a simulation environment allows one to directly determine impact of various human actions on the plant system. For example, actions that result in an overall degradation of the plant state could be considered erroneous, while non-normative actions that do not result in a degraded plant state would not be considered to be an error.

2.3.4 Discussion

Insight about the usefulness of these various error taxonomies can be gained by examining previous attempts to categorize human errors obtained from actual operating experiences. In the 1980s, the Electric Power Research Institute (EPRI) sponsored a series of nuclear plant simulator exercises intended to improve the quantification of human reliability [38]. In part, the study measured the crew's time-to-act following a stimulus (e.g., alarm). An objective of the study was to group the time-to-act results of the simulator exercises into three performance categories based on a skill-, rule-, and knowledge-based taxonomy. However, it was found that crew performance did not readily fit into a skill-, rule-, knowledge-based framework. Instead, the researchers identified three new performance categories:

- Type 1: a response following a change in plant state that is indicated by an alarm or value of a monitored parameter. In this case, crew response times fall within a probability distribution following the initial alarm.
- Type 2: a response following an event that gives rise to a primary cue that signals that action must be taken when a secondary parameter is exceeded or cannot be maintained below a threshold value. In this case, the crew response is delayed until the secondary parameter reaches the applicable limit.
- Type 3: a response following an event that gives rise to a primary cue that signals that action must be taken before a secondary parameter reaches a critical value. In this case, the crew's performance fell into two distinct categories – some crews immediately took anticipatory action in response to the primary cue while other crews waited until the secondary parameter neared the critical limit.

A conclusion of this study was that crew behavior was best characterized in terms of the cue structure provided by the plant rather than a cognitive framework. Furthermore, the behavior of nuclear operators cannot be easily categorized into a specific error taxonomy, particularly when the situational context is strongly driven by the informational structure of available cues. Rasmussen has cautioned that "human error is not a stable category of events and transfer of frequency data from one context to another depend on a subtle understanding of the psychological mechanisms and intimate intuition within both contexts." Rather than attempting to strictly define and quantify human error events, he stressed the need to create feedback loops around the human operator to make the effects of slips and mistakes observable and reversible [33].

Within the ADS-IDAC simulation environment, the distinction between errors of omission and commission has little meaning since both error types flow from the same cognitive model. Furthermore, the development of an information-driven decision-making model can readily accommodate the insights regarding cue structure found during the EPRI simulator studies. From this perspective, it is more important to clearly define the plant outcomes that constitute an error condition and then identify how the cognitive process led to the undesired outcome. Only then can the plant design, procedures, and operator training be improved to prevent or mitigate the undesired outcome.

2.4 Human Reliability Analysis Approaches

A key aspect of a probabilistic risk assessment is the identification and quantification of potential human error events for important operator actions. The identification of potential error events is generally based on some form of qualitative task analysis. The task analysis requires an analyst to decompose complex operator actions into smaller units that can be more easily analyzed. The task analysis (in addition to other supporting activities such as plant familiarization, operator interviews, and expert elicitations) will lead to identification of potential error events. Once error events are identified, a quantification method is used to estimate the probability that an operator will commit the error during the scenario of interest. Error quantification is typically based on the task type, various performance shaping factors that account for factors that influence operator behavior, possible dependencies between multiple actions that must be performed, and some consideration of time available to perform the action of interest. Human reliability methods generally all involve a qualitative task analysis followed by the application of a quantification method. However, the focus and decomposition level of the qualitative task analysis and the numerical methods used to quantify error events depend on the underlying human behavior model supporting the method. Looking at the historical evolution of HRA techniques, these methods can be categorized into three broad categories, or generations, based on their treatment of cognitive and contextual factors [1].

2.4.1 First Generation Approaches

Many of the first generation HRA models were based on either time-reliability curves or simple information processing models, such as a signal-organism-response model [8]. For example, the Technique for Human Error Rate Prediction (THERP)

[9] assigns nominal human error rates based on activity characteristics (including time availability) derived from the qualitative task analysis combined with performance shaping factors that account for some contextual factors. The level of task decomposition is more closely associated with the action type (e.g., operating a switch, closing a valve, executing a checklist) than the underlying cognitive processes driving operator behavior. Because these early first generation models had limited treatment of human cognitive processes, they are limited in their ability to fully account for contextual factors.

2.4.2 Second Generation Approaches

Second generation HRA models, such as the Cognitive Reliability and Error Analysis Method (CREAM), are built upon a stronger cognitive foundation than first generation methods [8]. Rather than assigning error probabilities based on functional task decomposition, the second generation methods generally decompose operator tasks into distinct cognitive activities (e.g., communication, diagnosis, monitoring) and human functions (e.g., observation, interpretation, planning, and execution). Human error rates are then determined based on the cognitive activity, human functions, and relevant performance shaping factors. Both the first and second generation HRA methods generally depend on a priori knowledge of the tasks that will be performed by the operator during an accident sequence. Because the scope of the task analysis supporting these methods is usually established by plant procedures, evaluating human error rates for actions outside the scope of the procedures remains challenging.

2.4.3 Simulation-Based Approaches (Third Generation)

Simulation-based approaches continue the evolutionary process from first and second generation human reliability approaches. In light of previous research efforts that have found that nuclear plant operators can be induced to commit unsafe actions under certain error forcing situational contexts [10], the ability to better model contextual factors should lead to improvement in human error prediction. Situational context includes factors such as the system state, the operator's state of mind, and the sequence and timing of events. The dynamic interaction and feedback between man and machine strongly affects the situational context and is difficult to model with a first or second generation HRA method.

Simulation-based HRA methods can address many of the shortcomings of earlier HRA methods. For example, the dynamic interaction and feedback resulting from operator actions can be directly modeled. The time-dependent behavior of performance influencing factors such as stress, fatigue, and work load can also be modeled within a simulation environment. Simulation-based methods can augment the data usually derived from time consuming and expensive control room simulator experiments conducted with actual control room crews. A computer simulation model can also explore a wider range of accident conditions than would be possible with actual control room operators in a simulator. An added benefit of simulation approaches is that a comparison between the simulation results and actual human performance on a similar task allows the simulation to benefit from feedback and develop greater accuracy and coverage [12]. Finally, a simulation approach allows a better determination of the consequences of a human error event.

Research related to the simulation of nuclear plant operator behavior has been active for the last two decades. During this time, a number of modeling approaches have been developed. In general, these approaches have been based on artificial intelligence methods, cognitive models, task analysis models, or stochastic error models. A survey of previous research efforts with significant implications for the ADS-IDAC approach is presented in the following sections.

2.4.3.1 Cognitive Event Simulator

The Cognitive Event Simulator (CES) was developed for the Nuclear Regulatory Commission in the early 1990s in order to explore the cognitive demands of problem solving in nuclear power plants [39]. The CES model uses a rule based artificial intelligence system capable of:

- Building and maintaining a coherent situational assessment under changing environmental conditions;
- Discriminating between expected and unexpected events based on the situational assessment;
- Performing a diagnostic search to evaluate possible hypotheses that would explain unexpected events; and
- Generating actions to respond to diagnosed faults.

The CES model did not attempt to model actual human cognitive processes, but instead focused on developing knowledge and reasoning rules capable of successfully coping with nuclear plant events. For example, the CES model had the capability to monitor at least 232 plant parameters every 10 seconds – far in excess of the capabilities of a human operator. It was assumed that the development of the artificial intelligence model would provide insights into the cognitive processes used by a human operator. The basic goal of the CES model was to identify a single cause for all observed abnormal indications. In comparisons with human simulator crews, it was noted that CES was able to develop a coherent hypothesis more quickly than human crews. Further, human crews tended to develop disjointed hypotheses for seemingly unrelated indications, while CES was able to rapidly "connect the dots" into a single diagnosis.

Although CES demonstrated good predictive capabilities in a limited number of situations, the goal of the ADS-IDAC project is to model a human operator, not a perfect decision maker. In order for CES to be a useful tool for identifying error forcing context, it would be beneficial if it was subject to the same vulnerabilities and shortcomings as a human operator. Furthermore, the reliance on a strict rule based approach appeared to become a significant detriment when attempting to extend the system capabilities into new areas. Therefore, CES approach was not used as a basis for the ADS-IDAC model, but instead served as an example of potential pitfalls to avoid. Because of the time needed to extend the capabilities of CES, the NRC terminated further CES development and instead shifted its research focus to empirical studies of nuclear operator behavior [25].

2.4.3.2 COSIMO

The Cognitive Simulation Model (COSIMO) was developed by the Commission of European Communities in the early 1990s to simulate the behavior of a nuclear plant operator during accidents [40]. The COSIMO cognitive model included four main functions: information filtering, diagnosing, hypothesis evaluation, and action execution. Information filtering was based upon physical salience and cognitive salience – only data that passed through these filters was perceived and used for further cognitive functions. The diagnosis function was performed by matching perceived data (i.e., data that had passed through the filtering process) to data stored in the operator's knowledge base. The matching operation was performed using similarity matching and frequency gambling. Each potential diagnosis was assigned a 'support' score based on its similarity to symptoms stored in the knowledge base and its relative base rate frequency. The diagnosis with the highest support score was then compared to an evaluation threshold based on the operator's cognitive state – if the diagnosis support score did not exceed the threshold value, the diagnosis process would be repeated. Once a diagnosis with a sufficiently high support score was identified, suitable follow up actions would be identified. COSIMO was structured to utilize action arising from either a rule-based frame or a knowledge-based frame. Rule-based frames are proceduralized actions while knowledge-based frames are intended to capture problem solving strategies based on heuristic rules and general engineering knowledge. COSIMO models three basic error types: cognitive collapse, unadapted change, and cognitive lock-up. Cognitive collapse is inability to reason clearly and is associated with information overload. Unadapted change occurs when an operator responds to new events without taking past events into proper account. Cognitive lock-up occurs when an operator continues to act in accordance with a hypothesis that should be revised or abandoned.

The COSIMO model was compared to actual human operator simulator experiments to determine how well the model reproduced human behavior [41]. The experiments used four nuclear plant operators to examine responses to five different events. During these experiments, two operator behaviors were identified that were not modeled in COSIMO – problem decomposition and timing of actions. It was observed that operators tended to use a strategy of decomposing the major event into a series of sub-problems. Furthermore, it was not necessary for an operator to diagnosis an event before applying a decomposition approach and stabilizing the reactor plant. Also, operators timed actions based on the rate of change of plant parameters and feedback from the reactor plant. These results suggested several areas where the COSIMO model could be improved:

- Capability to decompose problems into sub-problems;
- Ability to obtain plant feedback based on executed actions;
- Modeling of causal relationships between actions and plant responses;
- Capability to utilize an inferential strategy for problem solving.

The last capability was identified based on the observations that operators, even when under high time pressure, still used some limited inferential problem solving, rather than simple decision heuristics.

The COSIMO model experience has important implications for ADS-IDAC. Both the goals and basic architecture of COSIMO and ADS-IDAC are similar. The physical and cognitive salience based information filtering models in COSIMO could be easily modified and implemented within the ADS-IDAC framework. The use of a diagnostic method based on similarity matching and frequency gambling is very similar to the existing ADS-IDAC framework. ADS-IDAC also addresses some of the shortcomings of COSIMO, notably the ability to support problem decomposition into sub-problems. Interestingly, the experience with COSIMO underscores the need to model knowledge based problem solving strategies. Unfortunately, the knowledge based framework, which has direct applicability to the ADS-IDAC model, was never fully developed in the COSIMO model.

2.4.3.3 ACT-R

The Adaptive Control of Thought – Rational (ACT-R) model is a first principles approach to modeling human cognition and performance [42]. The ACT-R model includes two basic forms of knowledge: declarative memory for storing information chunks and production rule for implementing proceduralized tasks [43]. The ability to access the information stored in a declarative memory chunk is a function of its relative recency, the degree to which it is associated with the current situational context, a noise component, and a stochastic retrieval property. Production rules are activated by a similarity matching process that compares the current context to rules for implementing individual production rules. ACT-R includes five basic cognitive modeling paradigms:

- Instance Learning – As problems are solved, the problem solution is stored in memory as examples for future problem solving.
- Competing Strategies – Based on the past success of different problem solving strategies, the best (i.e., the strategy with highest utility) can be identified and used for future problem solving situations.
- Individual Differences – The model allows differences in individual cognitive ability (e.g., short term memory capacity, psycho-motor speed, or memory recall) to be simulated.
- Perceptual and Motor Processes – ACT-R interacts with an external environment through perceptual and motor processes that include the capability for modeling timing behavior, inputs, and outputs.
- Specialization of Task-Independent Cognitive Strategies – The ACT-R model can generate new production rules by combining existing rules through a learning process.

The use of production rules in ACT-R lends itself to modeling rule based errors such as the selection of inappropriate rules. The ACT-R model has found widespread use in modeling basic cognitive tasks such as puzzle problem solving [44]. A drawback of the ACT-R approach is that modeling has tended to focus on the "micro-cognition" of component tasks rather than full-scale dynamic skills [12]. However, recent efforts have attempted to apply ACT-R to more complex situations. For example, Byrne and Kirlik [42] recently used ACT-R to model runway incursion events for the National Aeronautics and Space Administration.

Although the architecture of ACT-R differs greatly from the ADS-IDAC framework, there are a number of modeling concepts that seem applicable to a nuclear plant operator cognitive model. Specifically, the use of declarative memory chunks for factual information and the use of production rules for proceduralized actions. Additionally, the activation and retrieval models used in the ACT-R context also have relevance for ADS-IDAC filtering and memory chunking operations.

2.4.3.4 ADAPT-MELCOR

The Analysis of Dynamic Accident Progression Trees (ADAPT) dynamic event tree approach was developed by Ohio State University in collaboration with Sandia National Laboratory. The ADAPT method is intended to facilitate the analysis of post-core damage scenarios (i.e., Level 2 PRA[1]) [45]. ADAPT has been linked to the MELCOR severe accident thermal hydraulic analysis code. MELCOR can model a wide range of severe accident phenomena including core degradation and

[1] Probabilistic Risk Assessment (PRA) studies are generally categorized into three distinct levels. Level 1 PRA refers to a risk study that spans the period from an event initiator to core damage. The result of a Level 1 PRA is the core damage frequency of a nuclear plant. A Level 2 PRA study bins the various Level 1 core damage results into distinct plant damage states and explores severe accident phenomenology (i.e., post core damage) such as core heat, melt, relocation, fission product transport, and containment performance. The result of a Level 2 PRA is a vector of fission product release categories and their respective frequency. A Level 3 PRA uses the fission product release categories to determine the health and safety consequences to the public by modeling factors such as meteorological conditions, population density, protective measure effectiveness (e.g., sheltering or evacuation), and radiological response. The results of a Level 3 PRA may include metrics such as prompt fatality or latent cancer rates.

relocation, containment performance, radionuclide transport and release, and hydrogen production. This approach has been applied to the automated development of Accident Progression Event Trees (APETs). APETs are used to analyze severe accident phenomenology and containment system response following a core damage event. Similar to ADS-IDAC and other dynamic event tree methods, ADAPT-MELCOR addresses sequence variability through the use of branching rules. For example, ADAPT-MELCOR has been used to analyze a station blackout scenario using phenomenological branching rules for creep rupture of major reactor coolant system components (e.g., pressurizer surge line, hot leg, and steam generator tubes), hydrogen combustion in containment, recovery of power, and relief valve failures.

A focus of the ADAPT-MELCOR research effort has been on improving efficiency and processing speed through parallelization of the computer code by spreading the computational effort over multiple computers. A method has also been developed using a "code-agnostic" approach to permit ADAPT to interface with different thermal-hydraulic engines, provided certain requirements are met. Despite these advances, ADAPT-MELCOR currently has very limited modeling of operator actions and cognitive decision-making processes. Furthermore, the simplified ADAPT-MELCOR plant model currently described in the literature does not appear to provide the full range of interactive controls necessary for an operator model to realistically simulate the plant operating procedures. Consequently, while this method represents a significant advancement in the state-of-knowledge of dynamic

PRA techniques (particularly in the Level 2 area), ADAPT-MELCOR has limited ability to model and predict errors of commission.

2.4.3.5 MCDET

The Monte Carlo Dynamic Event Tree (MCDET) [46] is a dynamic probabilistic risk assessment tool developed by the Gesellschaft für Anlagen-und Reaktorsicherheit (GRS), a non-profit research organization largely supported by the Federal Republic of Germany . MCDET is capable of combining stochastic behavior with a thermal hydraulic simulation code. The MCDET approach generates a dynamic event tree (similar to ADAPT-MELCOR and ADS-IDAC approaches), but uses a Monte Carlo simulation to model the stochastic behavior of timing parameters and system state changes. An initial step in an MCDET analysis is to divide the random parameters of a problem into two groups: (1) the generally continuous variables handled with Monte Carlo simulation, and (2) the remaining variables (generally discrete state transitions) modeled with event tree branching events. The actual analysis is performed in two steps. The first step generates sets of parameter values for the continuous variables using Monte Carlo simulation. In the second step, for each set of parameter values selected with Monte Carlo simulation, a discrete dynamic event tree is generated based on the possible combinations of discrete state transitions. This results in a collection of dynamic event trees with each tree representing a unique combination of continuous variables selected by Monte Carlo simulation and the branching events within each tree representing discrete state transitions given the selected values of the continuous variables. The MELCOR severe accident nuclear code is used as the thermal hydraulic model in MCDET, providing capability to simulate post-core damage scenarios. The MCDET approach has been applied to the analysis of a station blackout scenario for a pressurized water reactor [46].

Recent research activities have added a Crew Module to MCDET to permit the simulation of crew interactions with the plant model, including communications and performance shaping factors such as operator knowledge, ergonomics, and stress [47]. The crew model is based on a "script and routine" approach. Specifically, predefined operator tasks can be activated when certain prerequisite conditions are achieved (e.g., parameter values exceed a predefined threshold, component states, etc.). The plant information needed to activate scripts and routines is captured in a "key vector." While the timing of operator actions can be represented in the model, the current crew model does not simulate mental processes and the cognitive behavior of the operators. While the crew model has been used to analyze a secondary side feed and bleed [47] accident scenario, the current model has not focused on operator information perception, diagnostic, and decision-making processes.

2.4.3.6 ADS-IDAC

The theoretical foundation for ADS-IDAC has been under development for over a decade [23, 37, 48-50]. Early research efforts focused on developing necessary infrastructure to link a dynamic event tree scheduler driver with a thermal hydraulic simulation code capable of adequately representing a nuclear power plant. These early research efforts resulted in the development of the Accident Dynamics Simulator (ADS) approach which linked a thermal hydraulic representation of a nuclear power plant with a scheduling module capable of generating a dynamic event tree [51]. Parallel development of the Information, Decision, and Action (IDA) cognitive model established the theoretical basis for the model-based human reliability approach used in ADS-IDAC [23, 37, 48]. Later research efforts linked the ADS dynamic event tree scheduler and elements of the IDA cognitive model (extended to include crew interactions) to the RELAP thermal hydraulic nuclear plant analysis code [24]. The key distinctions between the ADS-IDAC current research effort and methods such as MCDET and ADAPT-MELCOR is the associated risk analysis domain and the treatment of human performance. Because the thermal hydraulic engine for ADS-IDAC is the RELAP code, the risk analysis domain is limited to Level 1 PRA studies (i.e., up to, but not beyond core damage). The use of the MELCOR severe accident code in MCDET and ADAPT-MELCOR allow these methods to explore post-core damage scenarios, including core melt and transport; containment performance; and fission product release. Adapting ADS-IDAC to use a severe accident code such as MELCOR, though feasible, is beyond the scope of this research effort. The application of parallel processing techniques, such as those used for ADAPT-MELCOR, can improve execution speed to increase the practicality of performing dynamic PRA studies. Although beyond the scope of this research effort, other researchers have developed and demonstrated parallel processing capabilities for ADS-IDAC [52].

Despite the maturation of the ADS-IDAC research program, several important limitations and needed enhancements were identified. Key areas focus areas for this research activity included improvements to the operator cognitive model, full development and implementation of a comprehensive operator knowledge base, and model validation. The foundations for these improvements are described in Section 2.5 and specific contributions of the current research effort are described in Section 2.6.

2.5 Cognitive Modeling Foundations

ADS-IDAC utilizes the IDAC cognitive model to drive predictions of human behavior. Although the basic structure of the IDAC cognitive model had been developed when this project was begun [22-24, 37, 50], a number of implementation details were missing from the model. For example, the specific model approaches for the following implementation aspects of the IDAC had not yet been included:

- information gathering and filtering;
- information assessment, event diagnosis, and situational assessment;
- modeling of skill- and rule-based actions; and
- the detailed structure of the decision-making model.

This section begins with a summary of relevant literature in the areas of information processing (Section 2.5.1), systematic biases in human decision-making (Section 2.5.2), and the foundations of naturalistic decision-making which provide the basic model for non-proceduralized actions in ADS-IDAC (Section 2.5.3). This literature highlights the background information necessary to develop the detailed models needed to fully implement the IDAC model

2.5.1 Information Filtering, Evaluation, and Utilization

ADS-IDAC is largely an information-driven modeling approach. Although the operator decision-making engine and associated knowledge base play a vital role in selecting appropriate operator responses for a given situation, an underlying assumption is that an operator responds to a perceived situation in predictable ways. Within this context, variations in the perception of information can drive behavior variability. Information filtering, assessment, and utilization all play an important role in capturing human performance variability.

2.5.1.1 Information Filtering

Most information-rich environments provide both useful data signals and distracting noise. Often, both signal and noise are distributed according to some random process and can overlap. The major challenge to a decision maker is filtering out noisy data in order to act only on genuine signals. One method of performing this filtering operation, derived from signal detection theory, is to establish a signal cutoff that establishes a binary decision threshold (see Figure 3) [53]. A parameter value that exceeds the cutoff threshold is considered to be a valid signal, while a value below the cutoff is considered to be noise. Unfortunately, a value above the cutoff value can be due to noise and a value below the cutoff can be due to an actual signal.

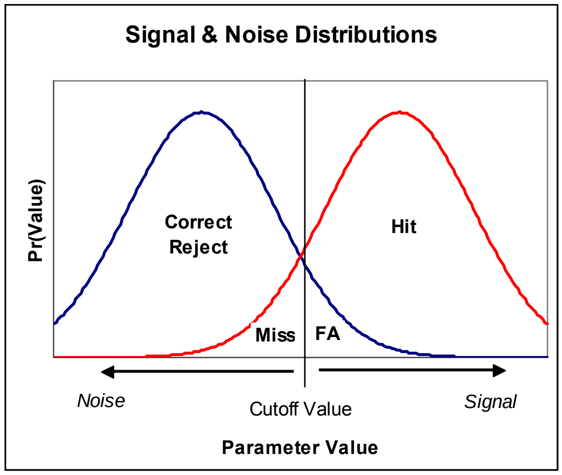


Figure 3 - Signal Noise Interaction

In general, four distinct outcomes could occur:

- Hit – the operator correctly identifies a true abnormal signal deviation and takes action;
- Correct Rejection – the operator correctly identifies the signal as within normal bounds and takes no action;
- False Alarm (FA) – the operator incorrectly interprets a noisy signal as a true deviation and takes action;
- Miss – the operator incorrectly dismisses an abnormal signal deviation as noise and takes no action.

The cutoff threshold establishes the relative ratio of hits, corrected rejection, false alarms, and misses within a specific environment. Various techniques have been proposed to find an optimum cutoff threshold that maximizes hits and correct rejections while minimizing misses and false alarms. For example, each of these outcomes can be assigned a utility value and, based on the probabilistic distributions of the noise and signal, an optimum cutoff value can be assigned that maximizes utility for the decision maker. However, research studies have indicated that decision makers, even after obtaining significant experience, often violate optimum cutoff strategies. To improve the agreement between signal detection theory and observed decision maker behavior, cutoff reinforcement learning (CRL) models have been proposed [54]. In general, CRL models assume that a decision maker will establish and adjust the cutoff value based on a learning process over a series of repeated trials. The learning process model includes factors such as positive reinforcement, generalization, and forgetting.

Barkan, Zohar, and Erev [55] compared four strategies for setting signal detection cutoff values for general industrial safety environments. The strategies included two static approaches based on utility and prospect theories, and two dynamic cutoff value approaches based on hill-climbing and cutoff reinforcement learning models. Each of these models included the notion that a "miss" did not always result in an accident – instead they assigned a conditional probability for an accident given that a miss event occurred. From the perspective of feedback and reinforcement, the probabilistic outcome for a miss introduces additional uncertainty since a decision maker cannot be sure if a non-accident outcome was due to a correct decision or a lucky near-miss. This study found that the cutoff reinforcement learning model captured several key behaviors, including initial tendencies toward risky behavior, slow learning toward optimum cutoff placement, and decreased learning speed as the conditional probability of an accident given a miss event is decreased. Because the static strategies do not allow adjustment of the cutoff value based on learning, the utility and prospect theory approaches do not model learning effects during repeated trials.

Application to ADS-IDAC Operator Model

Nuisance alarms are a potential source of operator distractions in the nuclear plant control room [56]. Nuisance alarms may be generated by maintenance and testing operations or by equipment failures. These alarms do not provide relevant information pertaining to the plant state and may distract operators from more important duties. A dynamic signal detection theory approach may provide a means to appropriately model operator behaviors in response to an alarm signal or an abnormal parameter measurement. For example, during maintenance or testing operations where nuisance alarms might be expected, an operator may “increase” their threshold to alarm conditions and be less likely to notice a true alarm. After maintenance and testing conditions are completed, the operators may not immediately return to their previous cutoff level and instead will need to ‘relearn’ an appropriate threshold. The identification and processing of nuisance alarms would also be expected to be a learning process, where an operator may pursue an initial alarm condition with a high level of diligence but follow-up on subsequent alarms with less vigor if the condition is perceived to represent noise. Additionally, training and experience would tend to establish nominal cutoff values for operators over time. New situations (such as accident conditions) or differing levels of experience among control room crew members may highlight problems with these established nominal cutoffs. The cutoff reinforcement learning model is beneficial for the construction of a dynamic parameter filtering function within the ADS-IDAC operator model.

2.5.1.2 Information Assessment and Utilization

In the mid-twentieth century, Egon Brunswik developed a lens model framework to explain certain processes in human judgment [57]. In his model, Brunswik separated internal psychological processes from the external world by a ‘lens’ composed of informational cues (see Figure 4). Each cue provides information about the true state of the external world in proportion to its ‘ecological validity.’ A cue’s ecological validity is analogous to the probability that the cue accurately predicts the state of the external world. The model assumes that an individual makes judgments about the state of the external world through a process of cue utilization. In essence, the Brunswikian Lens Model can be used to explain (at least in a qualitative sense) how judgments can become distorted from the true external world. For example, if the cue validities assumed by an individual do not agree with true ecological validities, or if the cues are utilized in a non-optimum manner, an individual’s judgments will be based on distorted information.

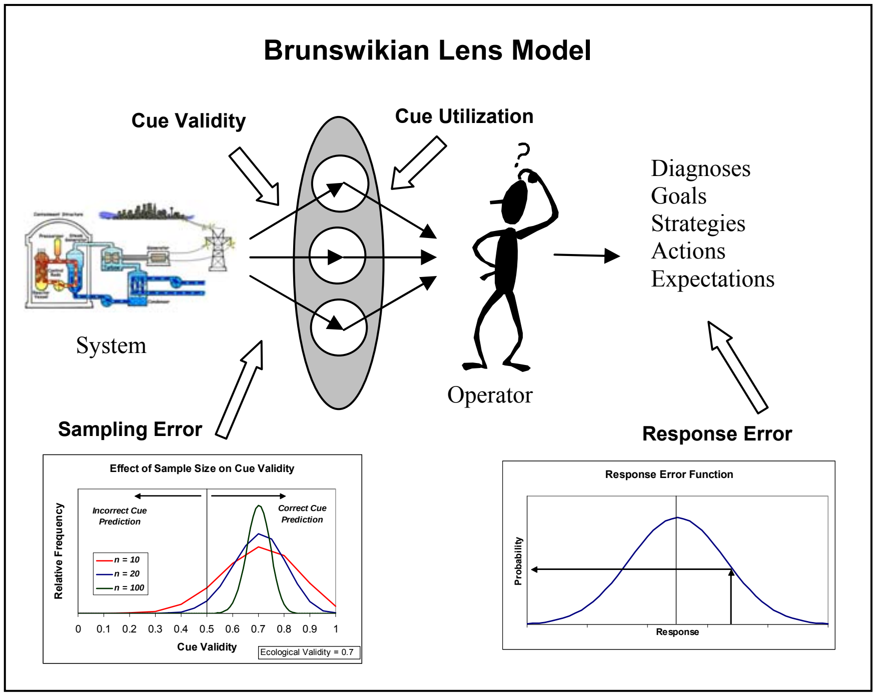


Figure 4 - Brunswikian Lens Model

Gigerenzer, Hoffrage, and Kleinbolting later modified the Brunswikian Lens Model and placed it inside a larger Probabilistic Mental Model (PMM) framework [58]. The PMM theory introduces two strategies for making immediate judgments: (1) use of a local mental model where judgments are made with certainty based on memory and logic, and (2) a probabilistic approach where a judgment is made based on probabilities obtained from the external world. The theory also introduces the concept of a target variable related to the decision basis for the judgment. For example, they noted that if an individual is asked "Which city has the larger population, San Diego or Birmingham?" the appropriate target variable is population. The PMM theory assumes that an individual will first attempt to make a judgment using a local mental model based on memorized information. However, if sufficient information cannot be retrieved from memory to make a judgment, the individual will resort to a probabilistic method. In this example, an individual might attempt to infer the relative populations of San Diego and Birmingham based on knowledge of the airports, professional sports teams, and key industries located in each city. The key difference between these strategies is that under a local mental model an individual attempts to retrieve information about the target variable directly from memory, while under the probabilistic approach the individual attempts to infer the target variable based on similarity of the problem to some reference class of objects. The inferred target variable value is based on the reference class and the internal "cue validities" used by the individual. The PMM theory uses the term "cue validity" rather than "ecological validity" to make the distinction between an individual's internalized experience with external world and the true "ecological validity." An individual is considered to be well adapted to the environment if his or her "cue validities" correspond to the true "ecological validity."

Juslin, Olsson, and Bjorkman [59] further extended PMM theory by adding sampling error and response error components. Sampling error accounts for differences between an individual's internal cue validities and the true ecological validity in the external world. Two sources of sampling error are proposed: (1) limited experience with the external world, and (2) formulation of internal validities using only a sub-set of representative cases stored in long-term memory. To account for differences between an individual's true confidence and reported confidence, the model also introduces a random response error. Using the combined sampling and response error model, Juslin, Olsson and Bjorkman predicted that over-confidence in decision-making will decrease with experience due to a reduction in sampling error.

A reduction in sampling error results in better correspondence between an individual's internal cue validities and the true ecological validity. Additionally, in environments with low predictability considerably more experience will be needed to achieve correspondence between internal cue validities and ecological validities.

Application to ADS-IDAC Operator Model

The extension of PMM theory by Juslin, Olsson and Bjorkman to include sampling and response errors has an important implication for the ADS-IDAC operator model. In the nuclear plant context, one could view cues as the symptoms of an abnormal event. For example, cues associated with a loss of reactor coolant accident might include decreasing reactor coolant pressure and high radioactivity levels in the containment building. To make an appropriate judgment about the meaning of these cues, an operator would need to rely on cue validities developed by past experience and training. Since the experience levels of operators might vary and training may not be directly representative of actual plant conditions, the operator's internal cue validities may differ from the true cue ecological validity. Also, operators may have different strategies for cue utilization which may result in different conclusions about a given plant state. Finally, since operators will generally communicate their beliefs and conclusions to other crew members, response errors could skew the perception of other members of the crew. In summary, the PMM theory with sampling and response error can provide a useful framework for modeling key aspects of nuclear plant operator behavior.

2.5.2 Heuristics and Biases in Cognitive Processes

The term “heuristics” refers to the relatively simple rules that people sometimes use to make judgments and decisions. According to Tversky and Kahneman [60], people rely on a limited number of heuristic principles which reduce the complex task of assessing probabilities and predicting values to simpler judgment operations. Tversky and Kahneman note that heuristics are economical and usually effective, but can sometimes lead to systematic and predictable errors. Gigerenzer, in making an analogy to visual illusions, notes that (1) our brain is not paralyzed by uncertainty when sufficient information is not available, (2) heuristic principles are used to make a “good bet”, and (3) this bet is based on the assumed structure of the environment [61] . Gigerenzer notes that heuristics are “not simply hobbled versions of optimal strategies” [62]. Indeed, he has placed heuristics within a broader decision making context of bounded rationality – where simple heuristics replace optimization strategies and the environmental structure is exploited to provide more adaptive, real world solutions.

In their 1974 paper, Tversky and Kahneman [60] introduced three main categories of heuristics: representativeness, availability, and adjustment and anchoring. Representativeness refers to similarity an object has with a larger class of objects. Tversky and Kahneman noted that individuals tend to make systematic errors when assessing if an object belongs to a certain class, notably the failure to properly account for the base-rate frequency of the class in the environment and the failure to adequately consider effects of sample size. Availability refers to the ease in which an individual is able to assess the frequency of occurrence of a class of objects.

Classes of objects that are more easily retrieved from memory will tend to be assessed as having a higher base-rate frequency than classes of objects that are more difficult to retrieve. If an individual does not have direct knowledge of events that can be retrieved from memory, he or she might attempt to construct a mental model to generate the needed frequency data. In this latter case, the ease with which a mental model is constructed will tend to increase the perceived base rate frequency of the object class. The creation of this mental model is closely related to the "simulation heuristic" [63].   When using a simulation heuristic, people construct a mental model to bridge an initial state with a target event by introducing a series of intermediate events. The target event is considered to be more probable if the intermediate events do not include exceptional or rare conditions. The simulation heuristic can also be demonstrated by evaluating "what if" scenarios for a given target event. In this later case, people will tend to restore any exceptional intermediate events back to a nominal state to break the casual chain of events in order to 'undo' a target event. Adjustment and anchoring refers to the process where people estimate a value by using available information to make adjustments to an initial starting value. Kahneman and Tversky noted that people tend to make insufficient adjustments from the starting value and therefore tend to bias their final estimates toward their starting point.

Tversky and Kahneman [60] noted that there are a number of contextual factors that can influence the systematic errors arising from heuristics. Framing issues, such as a more favorable description of an object, are likely to influence an individual judgment (e.g., a more favorable description may lead to a more optimistic prediction). Individuals may also develop inappropriate confidence in the validating of information (validity illusion). For example, an individual may exhibit a tendency to be more confident in predictions when given consistent data (even though the consistency might be due to dependencies among the data) than when predicting outcomes based on a more varied but statistically more significant set of data. It has also been noted that individuals tend to overestimate the probability of conjunctive events (i.e., the probability of a series of related outcomes) and underestimate the probability of disjunctive outcomes. This may result in people having unwarranted optimism about the success of an outcome, particularly when assessing complex systems in which a chain of events must succeed for overall success of the system. Anchoring and adjustment effects can affect an operator's expectations about how the system should respond to a set of actions. Consequently, the failure to accurately predict system response may result in an operator developing either an unwarranted pessimistic or optimistic view of the plant state.

The heuristics and biases literature has not been without controversy. With regard to context, Gigerenzer [64] has noted that the use of content-blind norms (i.e., application of a normative model for reasoning that does not consider context), is not appropriate for evaluating human judgment. Gigerenzer argues that the attribution of human behavior based on a heuristic and bias approach may not appropriately capture the influence of contextual information on human judgment. Although Gigerenzer referenced the classic conjunction fallacy associated with the "Linda Problem"[2] [65] in his discussion, he also provides an example in his 2005 paper illustrating the importance of contextual information [61]. In the 2005 paper, Gigerenzer relates the results of a research study where participants where asked to pour half the contents of a full glass into an empty glass. Participants were then asked to select which glass was "half full" or "half empty." Gigerenzer notes that participants relied on their prior knowledge of which glass was initially full when making their selection, thus demonstrating that eliminating contextual information from the problem (by equating a half full glass with a half empty glass) fails to capture the characteristics of human intelligence.

Wallsten [66] noted several situations where systematic errors attributed to heuristics may not strictly apply. For example, with regard to consideration of base-rates, he noted that expert may be more sensitive to base-rates compared to novice judges; the specificity of information provided by the problem may reduce the relevance of base-rate information; and the lack of salience of the base-rate information may reduce its weighting. Similarly, factors such as the sample to population ratio, problem wording, and experimental designs that fail to account for individual differences may explain the failure of experimental subjects to adequately

[2] The "Linda Problem" refers to an experimental study intended to explore the conjunction fallacy. Subjects are presented with a description of "Linda" that generally includes the following elements: "Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student she was deeply concerned with issues of discrimination and social justice, and also participated in antinuclear demonstrations". Subjects are then asked to select which of the following two alternatives is more probable: (1) Linda is a bank teller, or (2) Linda is a bank teller and is active in the feminist movement. Subjects often incorrectly choose alternative (2) even though alternative (1) is more likely. Kahneman and Tversky concluded that the availability and representativeness heuristics can make the conjunctive event appear to be more probable than the constituent events.

consider sample size. Wallsten's main conclusion was that the limitations associated with heuristics should not be overly generalized – rather a broader theory of heuristics should include the above factors.

Application to ADS-IDAC Operator Model

Given the time constraints for judgments and decisions associated with nuclear plant operations, it is likely that operators use some form of heuristics to make efficient decisions. For example, nuclear operators must continually assess emergent conditions to determine if they represent significant accident events or minor inconveniences. In essence, this problem is one of deciding the representativeness of abnormal events – is the alarm more representative of a major plant fault or a simple burned out fuse? In large part, this assessment is based on the operator's belief about the relative frequency of major accident events and more minor problems such as instrument failures. Because accident events are relatively rare, operators would probably not have memories of past events and would need to rely on a mental model based on knowledge and past training to place an abnormal symptom into a useful context. The contextual factors affecting operator decision-making may be strongly biased by experience and the memory of recent events. Once a mental model is developed, an operator might attempt to construct causal links leading to the perceived state. This process could bias the operator toward consideration of lengthy causal event chains containing more likely events than a shorter chain composed of rare events. This bias could be further reinforced by considering a combination of events that are consistent with the operator's simulation model to be more likely than the individual events themselves (similar to the conjunction fallacy). To the extent practical, the ADS-IDAC decision-making engine was constructed to accommodate these factors.

2.5.3 Naturalistic Decision Making

Naturalistic decision making (NDM) theory attempts to model how people make decisions in familiar, real-world contexts [67]. NDM is characterized several key elements [68]:

- Ill-structured problems;
- Uncertain, dynamic environments;
- Shifting, ill-defined, or competing goals;
- Action/feedback loops (a series of sequential decisions must be made, with early decisions affecting the context of later decisions);
- Time stress;
- High stakes; and,
- Multiple players.

The Recognition Primed Decision-Making (RPD) model is an NDM model that was formulated to explain how experienced firefighters identify and carry out a course of action without having to compare the merits of alternative actions [13]. This model was based in part on the observations that: (1) rarely did experienced fire ground commanders consider even two options concurrently, and (2) a search for an optimal choice could stall the decision maker long enough to result in loss of control of the operation [14]. In general, decisions are made within the RPD framework by a process of matching the current situation to a familiar or prototypical situation based on experience. Depending on the situational context and the decision-makers level of familiarity with the context, the RPD decision making process can take three different forms:

- Level 1 - Simple Match: The decision maker is able to recognize the current situation based on a feature matching process for relevant cues, goal states, expectancies, and typical actions. In this case, the decision process is simply a straightforward implementation of the typical actions for the recognized situation.

- Level 2 – Diagnose the Situation: The decision-maker is unable to obtain an adequate level of recognition for the current situation to enable use of the simple match strategy. In this case, a story building process may be used to augment or explain uncertain or missing cues. If the story building process results in adequate recognition of the situational context, the typical actions for situation can be implemented.

- Level 3 – Evaluate Course of Action: The decision maker performs a mental simulation of a possible course of action to determine if the action will result in undesirable consequences. If undesirable consequences are found, the decision maker will either modify the course of action or select a new course of action.

The RPD model has several key characteristics, notably: the first option generated by the decision maker is usually workable, the process relies on serial generation and evaluation of alternatives (rather than concurrent evaluation), the process does not attempt to find the optimal course of action, and the process focuses on situational assessment and mental simulation.

One controversial area for NDM models is defining what is meant by a decision error [69]. A working definition proposed by Lipshitz is that decision errors within a NDM model are “deviations from some standard decision process that increase the likelihood of bad outcomes.” An important feature of this definition is that bad outcomes can still occur as a result of sound decision process. Despite the possible difficulty in applying this definition, several researchers have examined factors that may increase the potential for poor outcomes within an NDM framework. For example, simulation experiments with naval combat information center crews found that the order that information was perceived by decision makers significantly influenced situational assessments associated with identifying an aircraft as civilian or military [70]. In particular, researchers noted that subjects tended to bias judgments in favor of more recent cues, even when earlier cues offered contradictory evidence. Another experiment examined the role of the “sunk-cost bias” and situational assessment in pilot decisions to continue visual flight into adverse weather [71]. The sunk-cost bias predicts that pilots would be more likely to continue a flight into adverse weather when the poor conditions were encountered near the end of a flight (i.e., the pilot has already invested a significant amount of time in the flight and would be biased to reach the desired goal). However, simulator experiments conducted with licensed pilots indicated that pilots were actually more likely to continue flights into adverse weather encountered earlier in the flight. This result was attributed to pilots placing more weight on the pre-flight weather briefing and less weight on actual weather observations early in the flight and increasing the weight of observed conditions later in the flight (as information from the pre-flight briefing became outdated). These studies underscore the influence of an accurate situational assessment on the eventual outcome of a decision.

Application to ADS-IDAC Operator Model

The RPD model has important implications for the ADS-IDAC cognitive model. The environments where RPD methods have been used are very similar to the nuclear plant control room. Specifically, nuclear operators routinely face uncertain, dynamic situations; have competing goals; time stress; and high stakes. The RPD model places greater emphasis on the operator’s situational assessment. Therefore, factors that influence information perception and evaluation can have a large influence on the potential for error events. A potential challenge of implementing the RPD model in ADS-IDAC is the need to formulate a mental simulation of possible alternatives when a simple match process is inadequate. This requires the development of a coherent mental model of the plant systems and the capability to extrapolate the consequence of proposed actions on future plant states.

2.6 Contributions of This Research to Human Reliability Analysis

This research project contributes to the fields of dynamic probabilistic risk assessment and human reliability in five main areas:

- enhancement to the modeling capabilities of the ADS dynamic risk assessment approach to support a wider variety of branching options, including stochastic timing variability, decision-making behaviors, procedure step-skipping.
- the development of an operator decision-making making engine capable of realistically representing control room operator performance
- demonstration of a full integration of a detailed human performance model with a realistic nuclear plant simulation;
- development of methods to organize and analyze human performance data within a model-based human reliability framework; and
- model calibration and validation through the use of actual nuclear plant operator performance data.

Each of these contributions is discussed in the remainder of this section.

2.6.1 Enhanced Dynamic Risk Assessment Branching Capability

ADS-IDAC generates a dynamic event tree during accident scenarios by activating branching points when certain conditions are met. Each branching point includes two or more individual event branches, each of which represents distinct combinations of system and operator states. Collectively, the branching points describe the topology of a discrete dynamic event tree (DDET) associated with an initiating event. A specific accident sequence is defined by the unique path through the DDET branching points from the initiating event to an end state. The generation of branching points is controlled by a set of general rules that define the specific activation conditions for a branching point. A key focus of this research effort has been extending the branching capabilities of ADS-IDAC to enable the realistic modeling of control room behaviors. This effort has resulted in the development of DDET branching rules capable of modeling stochastic timing variabilities among control room crews. Additionally, branching options have been developed to model variability in certain decision-making processes such as the activation and execution of skill-, and rule-based actions. Finally, enhanced capabilities to capture variability in crew goal and problem solving style selection and procedure execution have been added to the ADS-IDAC simulation code.

2.6.2 Development of Operator Decision-Making Engine

In order to more fully evaluate the role of contextual factors in influencing operator behaviors, it is necessary to develop a decision-making model capable of simulating the key types of decisions made by nuclear plant operators during accident scenarios. The decision-making model should have a number of characteristics, including a firm theoretical basis and the capability to represent features such as information gathering, perception and biasing; high-level goal and strategy selection; and static and dynamic influencing factors. Key elements of the decision-making process include the ability to support situational assessment through the use of a diagnostic module and the capability to link elements of the control room environment to the operator’s underlying mental representation of the nuclear plant systems based on knowledge and experience. The decision-making engine must also draw on an extensive knowledge base that represents not only proceduralized operator behaviors, but also catalogs significant skill- and rule-based activities. A key output of this research project has been the development of a decision-making engine that incorporates these elements. The decision-making engine responds to dynamic contextual factors and shows promise for advancing the state of knowledge in modeling and predicting a variety of human errors.

2.6.3 Full Integration of Human Performance and Plant Models

In order to realistically represent the contextual factors that can influence operator behavior, it is necessary to develop a detailed simulation of both the nuclear plant complex and the knowledge base used by operators to interact with the plant. A key focus of this research effort has been building a detailed model of a typical pressurized water reactor that includes all significant controls, indications, and alarms that the operators would normally rely on to address accident situations. The operator knowledge base was developed to accommodate a comprehensive suite of plant operating procedures, diagnostic information, non-proceduralized actions, and an underlying mental model of nuclear plant systems. In order adequately simulate the possible range of crew-to-crew variability, the needed infrastructure to support a variety of possible dynamic event tree branching rules was also developed. These branching rules can represent variations in timing, control input variations, activation of skill and rule based actions, information processing, and high level goal selection.

2.6.4 Human Performance Data Collection and Analysis

A model-based human performance analysis method such as IDAC provides a structured framework for the collection, storage, and analysis of human performance data. For example, the ADS-IDAC operator knowledge base provides a means to capture within a structured framework non-proceduralized operator actions; diagnostic information used by operators; and the linkage between plant components and the functions they support based on the operator's understanding of plant operation. A key insight from this research project is that operator performance data stored within the ADS-IDAC operator knowledge base repository can be readily be applied to new situations and problems. This provides a means to leverage limited and human performance experimental data collected with actual operating crews to a wider range of potential scenarios using a simulation approach. This can extend the usefulness of expensive experimental data to explore a wider range of operator performance variabilities.

2.6.5 Model Calibration and Validation

A challenge associated with the development of a complex model is that model parameters must be calibrated and some form of validation must be completed in order to establish the usefulness of the model. A significant portion of this research effort has involved using experimental data collected with actual nuclear plant operating crews to calibrate and validate the models used in the ADS-IDAC simulation code. The timing of this project has fortuitously coincided with an international human performance benchmark study that provided the opportunity to calibrate and validate ADS-IDAC against two sets of operator crew experiments performed at the Halden Reactor Project facilities in Halden, Norway. This research project not only demonstrates methods and approaches that can be used to calibrate a human performance model, but also establishes the usefulness of the ADS-IDAC approach in predicting potential sources of crew-to-crew variabilities.

3. ADS-IDAC Overview

This section provides an overview of the main components and features of the ADS-IDAC simulation code. In addition to a discussion of the architecture of the simulation code, also described are input requirements, simulation control, and output data.

3.1 Main ADS-IDAC Components

The ADS-IDAC model consists of three main components (Figure 5): the thermal hydraulic nuclear plant simulation model, the operator IDAC cognitive model, and the control panel which serves as the interface between the plant and operator models. The nuclear plant model is can be influenced by two factors – input

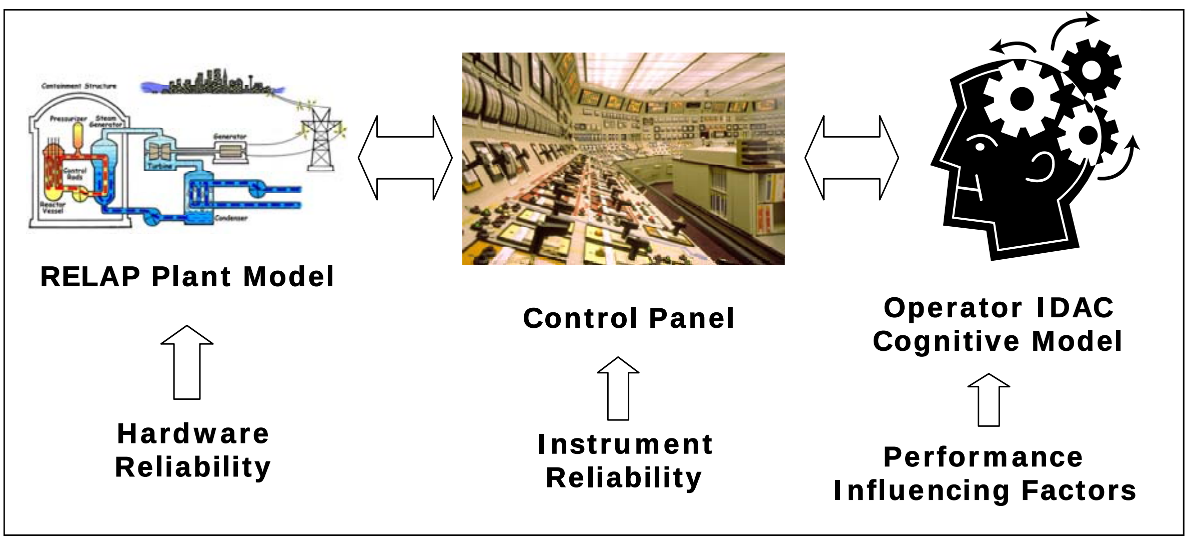


Figure 5 - Main Features of ADS-IDAC

from the control panel (e.g., manipulation of plant components) and activation of hardware failures. The operator model is influenced by information obtained from the control panel (and other members of the operating crew) and the set of static and dynamic performance influencing factors. The control panel, which serves as an information bridge between the operator and plant models, can be influenced by instrument failures that can cause bias or filter information received by the operator from the reactor plant. Information flow among each of these components is managed by the ADS scheduler. The schedule ensures information exchange among these components is appropriately synchronized, activates branching events based on the event tree branching rules, and ends sequences when termination criteria are met.

3.1.1 Nuclear Plant Thermal Hydraulic Model

The nuclear power plant thermal-hydraulic model in ADS-IDAC provides a rich contextual environment for the analysis and prediction of operator behaviors. The current version of ADS-IDAC utilizes the RELAP5/MOD 3.2 computer code [72] to provide a transient simulation of nuclear power plant operation. The RELAP5 code can simulate a wide variety of accident initiators and provides the capability to model key safety systems, controls, and instruments. Advantages of RELAP5 include its proven capabilities as a transient analysis tool and the availability of detailed power plant models. However, due to the intrinsic limitations of the RELAP5 code, it is not currently possible to model core damage states and severe accident scenarios. Consequently, ADS-IDAC is currently limited to the analysis of scenarios up to the start of core damage. Adaption of ADS-IDAC to a more versatile thermal-hydraulic engine, such as the TRACE or MELCOR code, has been identified as a future research activity.

Although RELAP5 plant models have been previously developed to support safety analyses and other regulatory uses, these models require some modification in order to exploit the full capabilities of ADS-IDAC. In general, ADS-IDAC requires the following modifications to an existing RELAP5 input model:

- Replacement of all conservative analysis assumptions with realistic best estimate parameters. These include trip setpoints, reactor power level, timing of automatic safety system actuations, and other key plant parameters.
- Modification to safety system models to replace simple boundary conditions with a more realistic representation of controls, instrumentation, and alarms. These modifications include modeling of redundant trains of multi-train systems, provisions for control of significant components such as key pumps and valves, and representation of critical support systems such as water supplies and electrical power.
- Addition of systems and components that provide a significant portion of the mitigative functions provided by the abnormal and emergency operating procedures.
- Implementation of interactive control interfaces for all significant components to allow the ADS-IDAC operator model to manipulate plant components. This includes the addition of a “manual” control mode for components that normally utilize an automatic control system (e.g., feed water regulating valves or power operated relief valves).

This research project has successfully integrated ADS-IDAC with a realistic three-loop, pressurized water reactor nuclear power plant RELAP model. The current plant model includes over 75 controls, 180 indicators, and 70 alarms. To improve feedback to the operator, the plant model includes reactivity and core power control features such as control rod movement, boration, and turbine load adjustment. Where necessary, controls for major pumps and valves in all front line safety systems (e.g., emergency core cooling and auxiliary feed water) were also added to the existing RELAP input model. As a result of these efforts, all major components referenced in the plant emergency procedures have been represented in the ADS-IDAC thermal-hydraulic model.

ADS-IDAC provides four possible control inputs for each component that can be manipulated by the operator: (1) changing the component operating mode (e.g., automatic vs. manual mode), (2) setting a specific control value for a component (e.g., throttling control valve to 50% open), (3) incrementing the control setting of a component (e.g., throttling open a control valve by an additional 10%), and (4) setting a control value based on a perceived parameter (e.g., setting the steam dump target pressure equal to the perceived main steam header pressure). These capabilities provide sufficient flexibility to realistically model all significant operator interactions with the plant model.

3.1.2 IDA Cognitive Model

The Information, Decision, and Action (IDA) cognitive model provides a framework for modeling individual operator behavior [23]. The basic elements of the IDA model are shown in Figure 6. As the name suggests, the IDA model consists

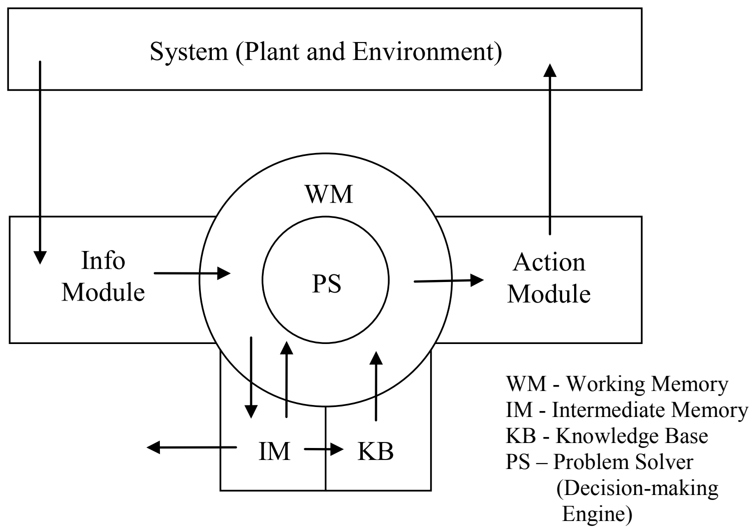


Figure 6 - Basic Elements of the IDA Model

of three main processes – information perception and processing, decision-making, and action execution. These cognitive processes are supported by a memory model consisting of three distinct units. Working memory stores recently perceived information, has a relative finite capacity, and is analogous to the operator's short term memory. Information residing in working memory is transferred to intermediate memory and is available for later retrieval. The capacity of the intermediate memory is assumed to be unlimited, but information may be forgotten (or decay) over time. The knowledge base is the preeminent repository of all information that the operators knows about the system, including procedures, facts, and past experiences. Within this memory model, intermediate memory serves as the buffer between working memory and the knowledge base. Over time, information stored in intermediate memory can be eventually transferred to the knowledge base during a learning process. Information can be retrieved from both intermediate memory and the knowledge into working memory when needed.

Operator behavior is influenced by static and dynamic performance influencing factors (PIFs) that capture internal and external factors that can affect cognitive performance [20]. The IDAC model groups PIFs into eleven broad categories:

- Cognitive modes and tendencies – alertness and attention
- Emotional arousal - stress
- Strains and feelings – task and time loading
- Perception and appraisal – situation perception and awareness of roles
- Intrinsic characteristics – confidence and motivation
- Memorized information – knowledge, experience, and skills
- Organizational factors – work practices and tools
- Team-related factors – cohesiveness, coordination, and leadership
- Conditioning events – latent hardware, software, and human failures
- Environmental factors – physical access, lighting, temperature, etc.
- Physical factors – fatigue, physical limitations

The status of the PIFs defines the operator’s mental state and influences the cognitive performance of the operator. Specifically, the operator’s mental state influences the information processing, decision-making, and action execution processes.

3.1.3 Crew Model

The ADS-IDAC crew model currently includes a decision-maker and an action-taker. The decision-maker is analogous to senior reactor operator (SRO) and the action-taker is equivalent to a reactor operator (RO). Similar to an actual control room, each operator has unique roles and responsibilities. The SRO selects the high level goal and directs all written plant procedures. The RO performs all interactions with the nuclear power plant model through the ADS-IDAC control panel. ADS-IDAC currently supports three high level goals: maintain normal operation, troubleshoot abnormal conditions, and mitigate accident conditions. Any of four problem solving strategies can be used to achieve these high level goals:

- Wait and Monitor – a passive information gathering strategy;
- Instinctive Response – perform simple skill based actions that are activated by matching perceived information to memorized situation-response profiles;
- Follow Written Procedures – implement formal written procedures (e.g., abnormal or emergency operating procedures);
- Knowledge-Based Reasoning – use a diagnostic process to guide crew actions in order to balance the flow of mass and energy within plant systems.

The selection of a specific goals and strategies is based on the plant information perceived by the operator and performance influencing factors. In general, the SRO selects an appropriate problem solving strategy for the crew based on the current high level goal and other factors. However, the SRO and RO may implement the Instinctive Response strategy whenever the perceived conditions match a memorized situation-response profile.

Each individual operator in ADS-IDAC is provided with profiling data that guide their behavior. The majority of the operator profile is devoted to the operator's knowledge base. The knowledge base includes rules for diagnosing plant events [16]; a functional decomposition and mapping of plant controls, indicators, and alarms; and rules for activating instinctive response actions. In addition to the knowledge base, the operator profile also includes data needed to: (1) calculate performance influencing factors; (2) define the operator's tendencies to skip procedure steps or pursue specific problem solving strategies; (3) manage memorized information; and (4) establish the timing of actions and communications. The flexibility afforded by the operator profile allows the simulation of a variety of operator performance tendencies. Specifically, performing influencing factors associated with problem solving styles, perception and appraisal of information, and utilization of memorized information can all be captured within the operator profile.

3.1.4 Scheduler and Simulation Control

ADS-IDAC generates a discrete dynamic event tree (DDET) to explore the impact of component failures and operator behaviors on plant safety. The DDET is constructed by allowing changes in plant and operator states at discrete points in time. Plant state changes include component actuations and failures while operator state changes may include decisions and interactions with plant hardware. This approach is categorized as an implicit state transition approach [73] and permits analyst-supplied rules to be used to direct plant and operator state changes. A main limitation of this approach is that the computational effort needed to obtain a solution exponentially grows as the number of modeled component and operator states increases. This exponential growth is known as sequence explosion and can limit the practicality of a simulation approaches.

The ADS-IDAC scheduler and simulation control module balances solution completeness with computational effort by focusing computational effort on certain sequences. During an ADS-IDAC simulation, component and operator state changes are permitted to occur at discrete branching points. State changes are modeled by generating one or more sequence branches at each branching point. Specific branching points and the number of branches generated at each branching point are defined by a set of analyst-supplied branching rules. Branching rules can be constructed to include sequence initiators, hardware and process variables, operators actions, and software (see Figure 7) [24]. A set of sequence termination rules are also identified to prevent excessive expansion of the DDET.

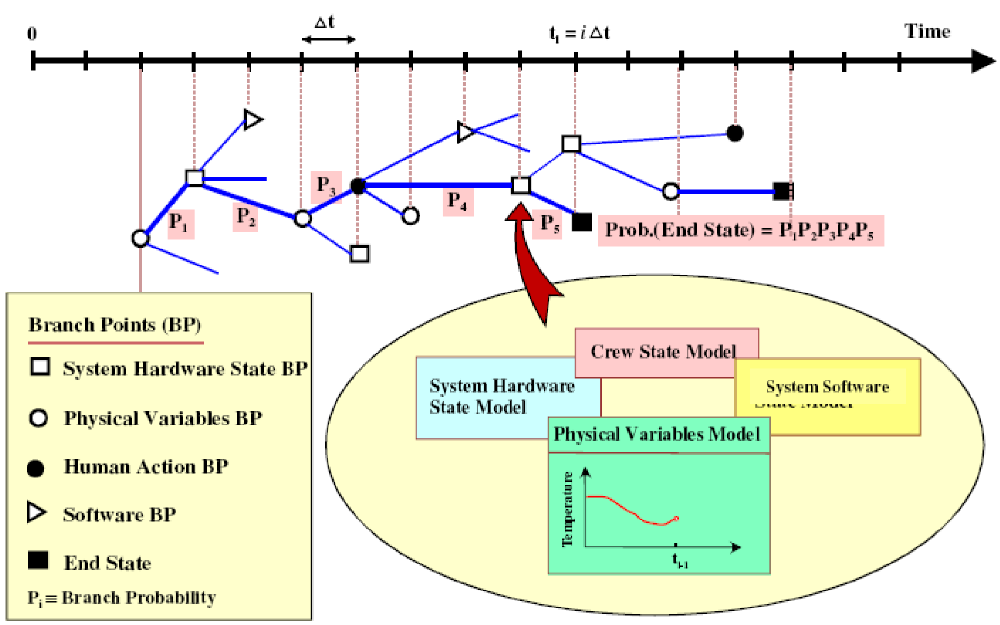


Figure 7 - ADS-IDAC Discrete Dynamic Event Tree

The process flow of the ADS-IDAC scheduling and simulation control functions are described in Figure 8. Scheduling control is handled by two nested timing loops. The inner loop handles execution of the RELAP thermal hydraulic plant model and is identified as blocks 4, 5, and 6 in Figure 8. The outer loop provides executive control over functions related to dynamic event tree generation, including initiation of initiating events, branching, and sequence termination. Consequently, execution of the ADS-IDAC code is controlled with two main time step parameters: (1) the internal RELAP5 time step (Δt_{RELAP}), and (2) the ADS-IDAC time step ($\Delta t_{ADS\text{-}IDAC}$). The RELAP5 time step establishes the incremental time step used in the reactor plant model thermal hydraulic calculation and is set low enough to ensure stable and accurate thermal-hydraulic modeling results. The ADS-IDAC time step establishes the incremental time step used by the ADS-IDAC scheduler module. Each ADS-IDAC time step, the scheduler pauses the execution of the RELAP5 thermal hydraulic code in order to update operator model data, activate hardware failures and initiating events, and manipulate controls. Between these ADS-IDAC scheduler time step pauses, RELAP5 runs without interruption using the time step control specified in the RELAP input deck. The inner loop RELAP time step is typically set to a maximum value of approximately 20 to 50 milliseconds, while the outer loop ADS-IDAC time step is set at least an order of magnitude higher at 0.5 to 1.0 seconds.

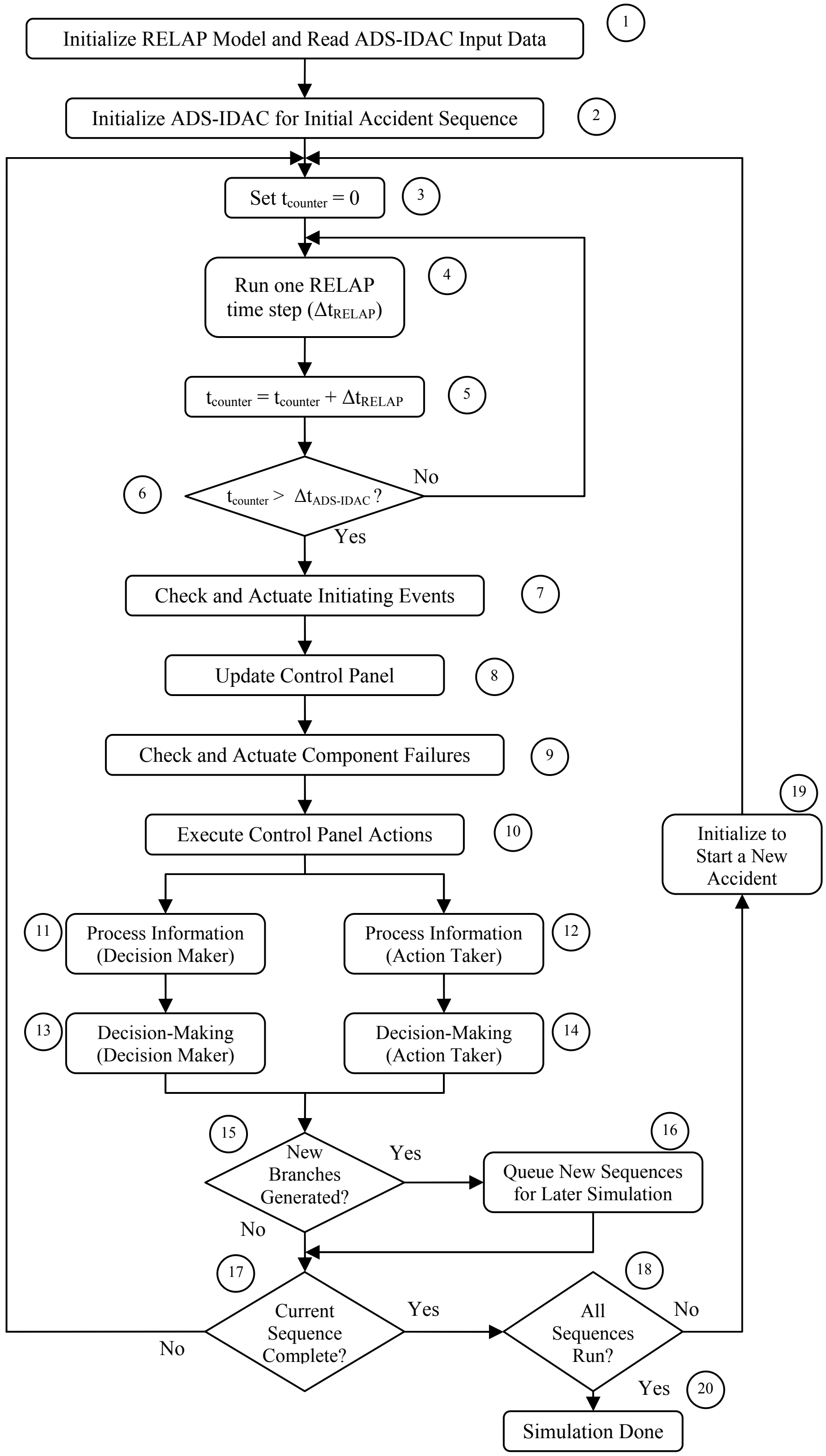


Figure 8 - ADS-IDAC Simulation Executive Control

The outer ADS-IDAC executive control loop performs the following functions:

- Activate initiating events (Block 7): ADS-IDAC can simulate two types of hardware failures: time based failures and conditional failures. Time-based failures are activated at a pre-determined time during an event sequence and generate only a failure branch (i.e., no success branch is initiated). Conditional failures are activated when a specific condition occurs (e.g., a component state) and generate both a success branch and a failure branch. Initiating events are typically modeled as time-based failures (e.g., an accident condition or transient is initiated at a specific time) while component failures during an event sequence (e.g., failure of mitigative equipment to function) are generally treated as a hardware failure.
- Update control panel information (Block 8): All indicators, alarms, and component states are updated based on the current RELAP thermal hydraulic data. However, simply updating control panel indicators does not automatically update the operator's perception of the plant state. The operator must first perceive and memorize information from the control panel in order to use the updated information.
- Activate hardware failures (Block 9): Component failures activated by a change in component states are activated by generating a success branch and a failure branch.

- Execute operator interactions with the thermal-hydraulic plant model (Block 10): The crew interacts with the thermal-hydraulic model through the use of RELAP interactive controls. A crew action changes the value of the associated interactive control in the RELAP thermal-hydraulic model; when RELAP is restarted in Block 4, the result of the interactive variable change can influence the reactor plant simulation. This step performs the action step of the IDAC cognitive model. The actions arising from the information processing decision-making process from the previous outer loop time step (i.e., the actions arising from the crew decision making processes in Blocks 13 and 14 are executed during the subsequent time step in Block 10).
- Perform information processing and decision-making processes for each crew member (Blocks 11 – 14): These steps implement the information and decision-making processing of the IDAC cognitive model. Actions arising form these process steps are executed in subsequent outer loop time steps in Block 10.
- Queue new dynamic event tree branches (Block 16) When a new sequence is identified, the ADS-IDAC scheduler stores all information related to the current plant and operator for later use. Information related to placement of the new sequence within the simulation Discrete Dynamic Event Tree is also saved (e.g., identity of parent sequence branch, number of new sequences generated, and events to be partitioned among the new sequences). New sequences are then placed in a holding queue to await later simulation.
- Terminate sequences (Block 17), activate new dynamic event tree branches (Block 19), and terminate the overall simulation (Block 18). The purpose of these

steps is to determine if any termination criteria have been met by the current sequence. In general, three criteria can be used to terminate a sequence: (1) the simulation time has reached the specified truncation time, (2) the sequence reached the specified probability cutoff value, or (3) the sequence was terminated by a special procedure following command or the activation of a special alarm condition. The later condition allows the analyst to stop a sequence when a specific condition or parameter value is reached.

A detailed description of the ADS-IDAC simulation executive and simulation control is included in Appendix K.

3.2 ADS-IDAC Input Requirements

ADS-IDAC requires a significant amount of input information in order to define control panel instrumentation, controls, and alarms; populate the operator knowledge base; identify accident initiating events and plant hardware failures; and provide branching rules and sequence termination criteria needed to construct the dynamic event tree. Although creation of an initial suite of input files can be resource intensive, much of the input data is generically applicable to a wide range of possible scenarios. For examples, control panel descriptions, procedure content, and sequence termination criteria can be generically applied to a wide spectrum of abnormal and accident scenarios. Appendix K provides a detailed description of all ADS-IDAC input requirements.

3.2.1 Control Panel

All plant status information perceived by the operations crew must be displayed on the ADS-IDAC control panel and all control manipulations must be performed through the control panel interface. This control scheme is similar to an actual control room, with the ADS-IDAC control panel serving as the main information interface between the operators and the reactor plant model. Three main categories of information can be displayed on the ADS-IDAC control panel: (1) reactor plant thermal hydraulic parameters (e.g., temperature, pressure, flow rate), (2) component operating state (e.g., on, off, open, closed), and (3) alarms. Indicators for thermal hydraulic parameters can display both the value of an indicator or the rate of change of the target parameter. The rate of change of a parameter can be used to provide a trend display for use by the simulated operators (similar to a strip chart recorder). Component operating state information can be used to model simple panel status lights (e.g., pump operating status). Finally, alarms based on parameter values, component operating state, or the difference between two parameters can be displayed on the control panel.

Operators may manipulate active components in the RELAP model using two types of controllers: (1) continuous variable control with fine adjustment capability (control values can be assigned over a range of acceptable values), and (2) discrete controllers that permit simple component state changes (e.g., open, close, off, or on). Continuous variable controls are used for components such as throttle valves and setpoint controllers to increase the realism of control interactions with the plant model. Discrete controllers are used for components with a small number of operating states such as pumps or control switches.

The plant control panel input files establish the linkage between the RELAP thermal hydraulic model and the ADS-IDAC operator model. All interactions with the thermal hydraulic model involving active components (e.g., pumps, valves, and setpoint controllers) utilize the interactive variable feature of RELAP. Input data is needed to describe the specific interactive variables that will serve as communication channels to link the RELAP plant model to specific control panel controls. For example, the panel controls used to stop and start the emergency core cooling system are linked to specific interactive variables in the RELAP plant model. Depending on the value of the interactive variable, the thermal hydraulic model can change the state of the associated pump or valve component.   All other information exchange such as parameter values and component states is communicated by linking the ADS-IDAC control panel to the associated control variable, hydrodynamic component, or logical flag within the RELAP model. In general, temperature, pressure, and flow data is communicated by linking ADS-IDAC to the associated hydrodynamic component and parameter of interest in the RELAP model. Values that must be calculated such as subcooling margin, average temperatures, and liquid levels, are generally handled with control variables. In these cases, the ADS-IDAC control panel is linked to the associated control variable in the RELAP model. Some alarms and safety system actuations are modeled with logical trip variables; in these cases, ADS-IDAC is linked to the associated Boolean variable in the RELAP model. A sample listing of the ADS-IDAC control panel for a three loop pressurized water reactor is provided in Appendix B.

3.2.2 Operator Knowledge Base

Each individual operator in ADS-IDAC is provided with unique knowledge base and operator profile to guide the operator's behavior. The knowledge base includes rules for diagnosing plant events; a functional decomposition and mapping of plant controls, indicators, and alarms; and rules for activating instinctive response actions. In addition to the knowledge base, the operator profile also includes key parameters for calculating performance influencing factors; defining the operator's tendencies to skip procedure steps or pursue specific problem solving strategies; managing memorized information; and timing of actions and crew interactions. The operator profile contains information pertaining to the calculation of performance influencing factors, problem solving styles, perception and appraisal of information, and utilization of memorized information. Section 4.3 provides additional detail on the operator knowledge base and Chapter 7 provides an overview of performance influencing factors in the ADS-IDAC model.

3.2.3 Plant Hardware Failures

ADS-IDAC allows the analyst to model two types of hardware failure events: (1) time dependent failures, and (2) conditional demand failures. Time dependent failures allow the analyst to initiate hardware state changes, including failures, at a prescribed time during the simulation. Time dependent failures can include hardware failure events, such as failure of pump or a turbine or reactor trip, or may also activate accident initiators included in the RELAP thermal hydraulic model such as losses of reactor coolant or steam line breaks. Conditional failures are triggered when a specified component changes operating state. For example, activation of the reactor trip alarm can be used to generate a conditional failure of an auxiliary feedwater pump. Time dependent failures generate only a single failure event sequence branch while conditional failures generate two event sequence branches – a success path and a failure path. However, both failure types permit the operators to attempt to recover the failed equipment. If the operator attempts to recover failed equipment, additional sequence branches representing component recovery and permanent failure are generated. Thus, each conditional failure event can result in three outcomes: (1) the equipment does not fail, (2) the equipment initially fails but is later recovered, and (3) the equipment fails and is unrecoverable. Time dependent failures result in two possible outcomes: (1) the equipment initially fails but is later recovered, and (2) the equipment fails and is unrecoverable.

Appropriate input data must be provided by the analyst to describe how ADS-IDAC handles component failures. All failure events must by linked to an active component controller on the ADS-IDAC control panel. For time dependent initiating events associated with passive component failures (e.g., piping system leaks and ruptures), this generally requires that an artificial controller be set up on the ADS-IDAC control to activate the accident initiators[3]. For all failures, the analyst must

---

[3] Accident initiators that are not related to active component failures are typically modeled in RELAP by inserting a valve component into the system that provides a vent path to the environment or an

specify the associated control panel controller, the desired failure state, the failure probability, and the recovery probability. For time-dependent failures, the failure time is specified; for conditional failures, the component that triggers the failure event is specified. A detailed discussion of the input requirements is provided in Appendix K.

3.2.4 Dynamic Event Tree Control

In order to achieve complete scenario coverage using dynamic probabilistic risk assessment methods, it is necessary to explore a large number of accident sequences. New accident sequences are generated when a branching rule is activated; this leads to the generation of two or more distinct sequences depending on the number of system or operator states specified by the associated branching rule. ADS-IDAC can generate event sequence branches for a wide range of events. The current version permits the generation of branches based on the following elements:

- Component failures and recovery (discussed in Section 3.2.3);
- Time required to initiate the actions associated with a mental belief (i.e., activate skill- or rule-based behaviors) (new feature added by this research);
- Stochastic time variability required to perform a specific control panel action (new feature added by this research);
- Control input variations when executing control panel actions (new feature added by this research);

---

adjacent system. For example, a steam generator tube rupture is modeled by inserting a normally closed artificial valve between the primary and secondary sides of a steam generator tube. When a steam generator tube leak is initiated, the valve is opened to a position commensurate to the desired leakage rate. When a failure is initiated in this manner, operator recovery to terminate the leakage by shutting the artificial valve is not possible.

- Omission (skipping) a proceduralized or skill- or rule-based action (new feature added by this research).

The analyst must specify the branching rules that activate these various branching events. Hardware failure branching rules identify the affected components, the activation conditions, and the failure and recovery probabilities. For timing related branching events, the analyst must identify the specific action or mental belief that will trigger branch generation and the number of branches that are generated. The conditional probability for each time branch is calculated from the timing distribution specified for the associated action or mental belief. Similarly, control input variation branching rules specify the associated action and a probability table that specifies a discrete control input value and its associated probability. An error of omission (i.e., step-skipping) is modeled as a binary branching event – either the step is performed or it is omitted. Step-skipping is not activated by branching rule; instead the analyst specifies a skipping probability cutoff value, and if the calculated step-skipping probability is higher than the cutoff value, a step-skipping branching event is generated. This allows the analyst to suppress lower probability skipping events to prevent the generation of excessive numbers of event tree branches during a simulation (i.e., sequence explosion).

Simulating a large number of sequences often requires a significant amount of computational power and time. Therefore, uninteresting sequences are often terminated or truncated. ADS-IDAC provides several methods to terminate or truncate accident sequences. These truncation methods are based on sequence elapsed time, sequence probability, and conditional events. Branching rules and sequence termination criteria are described in greater detail in Section 8.3 and the specific input requirements are provided in Appendix K.

3.3 ADS-IDAC Output

ADS-IDAC generates a number of output data files that describe the dynamic event tree sequences to allow the analyst to identify conditions leading to a degradation in the safety of the nuclear plant. Since a goal of the ADS-IDAC research project is to provide a tool to study the influence of context on operator behavior, sufficient output information is provided to fully characterize the contextual factors. Three general categories of output files are generated by ADS-IDAC: dynamic event tree information, crew information and performance influencing factors, and thermal hydraulic data. The output files are all written in plain text file format and can be imported into a third-party program such as MS Excel for data analysis and visualization. A detailed description of the ADS-IDAC output files is provided in Appendix K.

3.3.1 Discrete Dynamic Event Tree

The ADS-IDAC output provides all information required to construct a graphical representation of the dynamic event tree. Specific output for each branching event includes:

- type of branching event and number of branches generated;
- the time of the branching event was activated; and

- the sequence identifiers associated with the branching event.

Although earlier versions of ADS-IDAC included sufficient post-processing capability to graphically generate the event tree, this capability is no longer functional. This capability can be added into a future update of the ADS-IDAC code, but is beyond the scope of the current research effort.

In addition to the data needed to reconstruct the dynamic event tree, ADS-IDAC also provides a sequence summary list and a complete narrative description of each sequence. The sequence summary lists all event sequences generated during the simulation along with the associated sequence probabilities, termination times, and reasons the sequences were ended. The sequence narrative provides a time history of all control room alarms, crew communication, branching events, and operator interactions with the control panel. A sample sequence narrative for an uncomplicated reactor trip scenario is provided in Appendix E.

3.3.2 Crew Behavior and Influencing Factors

Crew related output data includes the state of all dynamic performance influencing factors, diagnostic output, information gathering information, and the output from the procedure step-skipping module. Dynamic performance influencing factor information includes a time history of each dynamic PIF (system criticality, time constrained loading, and information loading) for each operator including the values of he parameters supporting calculation of the PIF value. Diagnostic information summarizes the time history of the relationship values for all events include in the diagnostic matrix for each operator.

The ADS-IDAC information processing model includes a control panel scanning feature for active information gathering (see Section 4.1.2.2). During each scan cycle, ADS-IDAC outputs a listing of each parameter and its associated priority level included in the operators scan queue. This information includes the following elements:

- the time of the scan cycle;
- identification of the associated operator;
- the total size of the associated operator's scan queue (i.e., the total number of control panel items scanned); and
- the contents of the operator's scan queue (i.e., specific alarms, components, and parameters included in the control panel scan).

Because each operator can have a unique knowledge base and may perceive different information during the course of an accident scenario, PIF values, diagnostic, and control panel scanning results for each member of the operating crew are not identical.

Because the operators do not execute procedure steps simultaneously, procedure step-skipping information for each operator is combined into a single output data file. The data output summarizes data associated with the ADS-IDAC procedure step-skipping module and includes the following information:

- sequence time;
- the associated operator;
- the procedure name and step number;
- the associated action;

- the relevance value of the action to operator's current situational assessment;
- other dynamic factors associated with step-skipping model including the time constraint loading for the operator;
- static factors associated with the procedure step; and
- calculated probability of skipping associated step action (i.e., committing an error of omission).

The crew behavior and influencing factors output information provides a complete time history of the factors associated with the operator's mental state and the impact the mental state has on operator behavior.

3.3.3 Thermal Hydraulic Data

The ADS-IDAC definition of a human error event is based on the impact human activities have on meeting a plant's functional needs. Therefore, in order to determine if a specific set of operator behaviors constitute an error event, it is necessary to fully understand the impact operator actions have on plant thermal hydraulic processes. ADS-IDAC provides a sequence-specific comprehensive time history of all control panel parameter values; component and alarm states; and controller input values. This allows the analyst to review the time history of plant parameters to determine if safety margins were compromised or if safety limits were exceeded as a result of the operator activities during a specific sequence. Although earlier versions of ADS-IDAC included limited graphical capabilities to view key parameter histories, this capability no longer exists. This feature can be added into a future update to the ADS-IDAC code, but is beyond the scope of the current research effort.

4. Information Processing and Knowledge Base Modules

A key feature of the ADS-IDAC simulation model is that all operator behaviors arise from perceived data rather than the direct output from the plant thermal-hydraulic plant model. Before an operator can use any plant information, the data must first pass through the operator's perception filter. Because the perception filter can either screen out or distort data obtained from the plant model, the operator may possess incomplete or inaccurate information. Furthermore, the cognitive processes for each operator are supported by an individualized knowledge base. The knowledge base represents the memorized information, skills, experience, and abilities available to the operator. If the operator uses this incomplete or inaccurate data to guide their decisions and actions, human error events may occur. A goal of this research is to develop necessary modeling and simulation capabilities in ADS-IDAC to identify situations and contexts where operators may implement inappropriate actions due to limitations of the information perception process or the operator knowledge base.

4.1 Information Collection

A key feature of the ADS-IDAC information collection process is that the operator's attention is generally directed to data that the operator perceives as most relevant to the current plant state. For example, if the operator has perceived a problem with the feed water system, attention generally will be focused on relevant parameters such as steam generator water levels and feed water flow rates. Because the operator does not possess infinite information processing capabilities, shifting focus to a new area will result in reduced attention to other areas. Therefore the dynamic information collection model causes the operator to spotlight information perceived to be relevant and to ignore information deemed less important. If the operator has developed an accurate situational assessment, this process serves to improve the efficiency of limited information processing capabilities. However, if the operator's situational assessment is incorrect, the operator is more likely to miss important information and be less likely to mitigate an accident event.

Information drives all important operator behaviors in the IDAC model and influences goal and strategy selection, formulation of the operator's event diagnosis and situational assessment, the activation of non-proceduralized actions, and the verification of the impact of recent operator actions (see Figure 9). Observations of nuclear plant control room activities have determined that operators actively and passively gather information from a variety of sources [74]. Active information gathering occurs when the operators specifically seek information about the status of a parameter, alarm, or component state. Active information collection generally refers to two main types of information collection – information gathering directed by procedures and periodic control panel scanning. Since the operator intends to gather and use actively collected information, this information would have a lower likelihood of being filtered. Passive information gathering occurs when the operator receives unanticipated information from another crew member or perceives the actuation of an alarm. Because passively gathered information is not intentionally collected and may not be pertinent to the operator's perceived plant state,

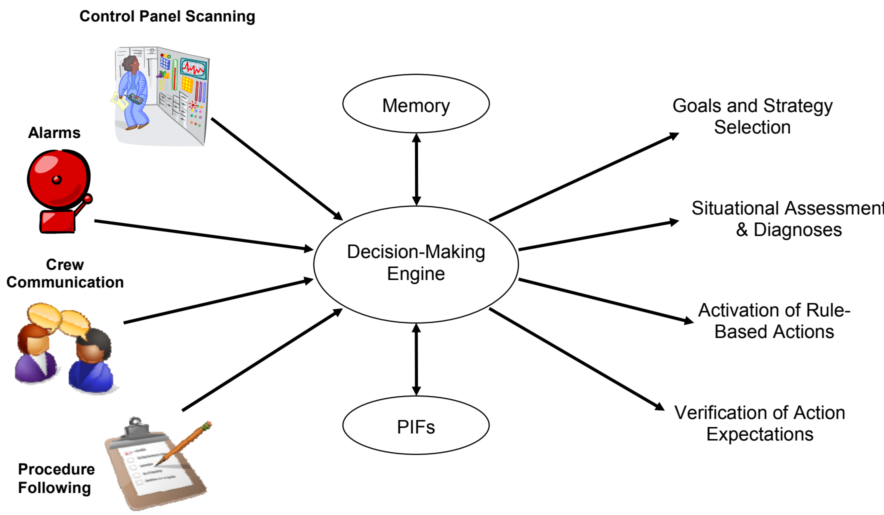


Figure 9 – Information Sources and Uses

it has an increased likelihood of being bypassed or missed. All information, regardless of whether is collected actively or passively, is subject to biasing.

All perceived information is stored in a memory repository for later use. Information that is available but not perceived is not placed in memory and is not available for later use. When the operator requires information, the memory is searched to determine if the information has already been perceived. If the information is stored in memory, the operator may use the stored data depending on the preferences and tendencies established in the operator's profile (i.e., does the operator tend to rely on information in memory when available, or is updated information always sought). Using information already stored in memory can be more efficient for the crew because it is readily available and is not subject to further biasing, filtering, or communication errors. However, depending on the recency of the information, it may no longer accurately represent actual plant conditions.

4.1.1 ADS-IDAC Information Processing

The ADS-IDAC information processing module is shown in Figure 10. The information processing stage of the IDAC cognitive model consists of three main steps: (1) gathering information, (2) sorting information based on its origin and the type of information, and (3) initial information processing. Information being actively processed during this stage can be considered to be held in the equivalent of short term memory. At the conclusion of the information processing stage, all information (including any biasing factors) is transferred to the operator's intermediate memory[4].

ADS-IDAC employs a central communication clearinghouse (or "blackboard") architecture for handling all information interchange. All communication items (or communication packets) initiated by members of the operations crew and the control panel are posted to the central blackboard. Each communication packet identifies the sender, intended recipient, type of communication, and the content of the message. For the purposes of ensuring adequate control over communication processes, the control panel is considered to be

[4] Operator memory in ADS-IDAC consists of three areas: short term memory, intermediate memory, and long term memory. Short term memory is used to temporarily store information that has just been collected. Once data in short term memory is perceived, it is stored in intermediate memory. Information in intermediate memory can be used to support decision-making, but may decay over time. Long term memory is static and consists of the factual information the operator knows about the reactor plant and procedures.

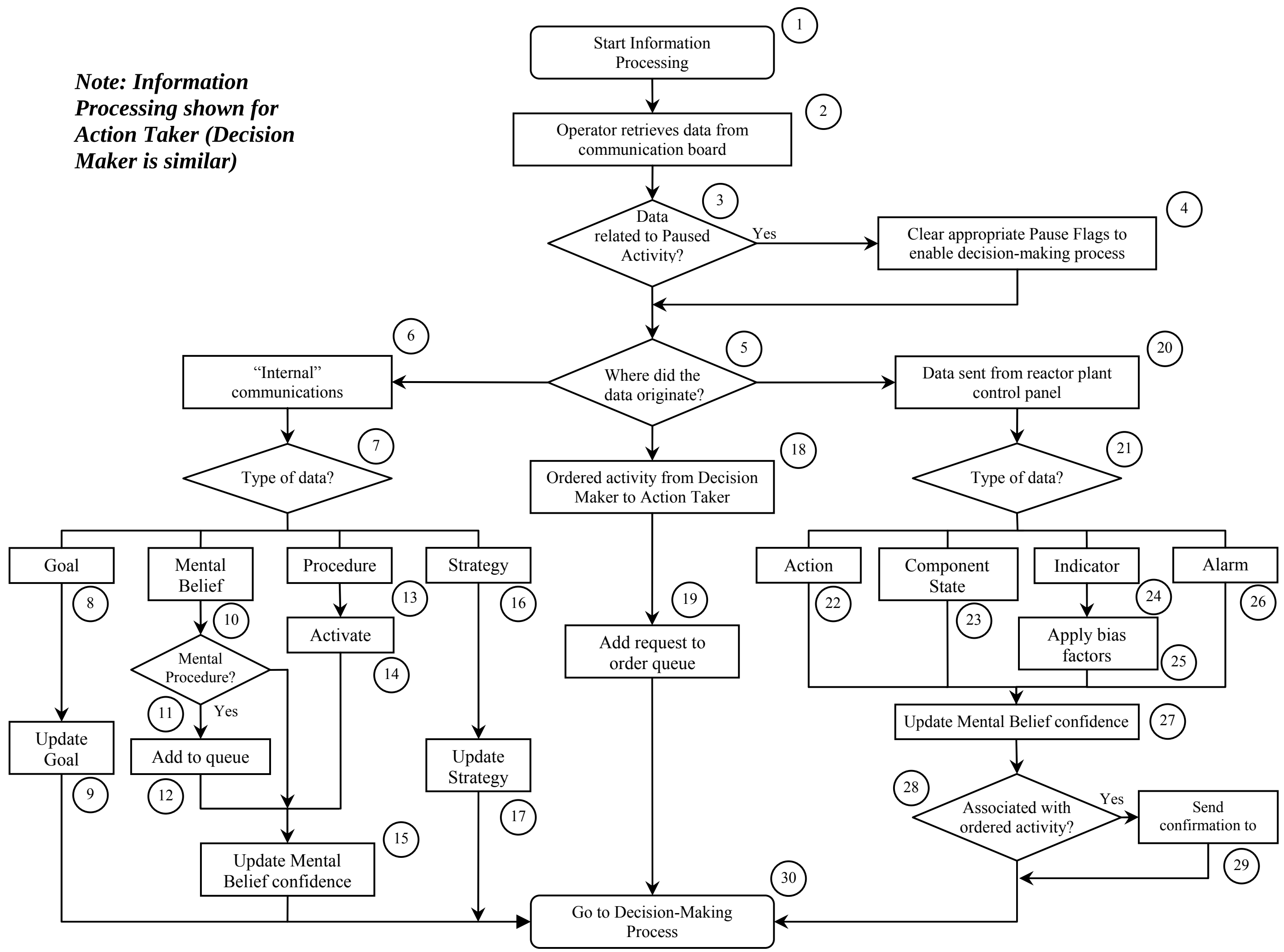


Figure 10 - ADS-IDAC Information Processing

a model agent equivalent to an operator. For example, the reactor plant might post the actuation of an alarm to the blackboard with all operators identified as intended recipients. Similarly, when the reactor operator attempts to open a valve or start a pump, a message for the reactor plant is posted to the blackboard. The information processing stage begins with each control room operator and the reactor plant control panel scanning the communication blackboard to identify any messages listing them as the intended recipient (Block 2). All communication blackboard information that is addressed to the operator (and has not been designated as being withheld from the operator) is loaded into that operator's short term memory.

After information is obtained from the communication blackboard, a series of sorting steps are used to categorize the new information (Blocks 5, 7, and 21). Information can originate from three sources: internally from "within" the operator and externally from the control panel or from another operator. Internal information transfers information related to cognitive decisions within the operator's own memory. This internal information refers to distinct decisions made by an operator and includes changes in high level goals, new problem solving strategies, execution of a new procedure step, and new mental beliefs. External information from the control panel includes the status of control actions, component states, parameter valves, and alarm information. Information can also be transmitted among the control room crew members. In the current implementation of ADS-IDAC, only the reactor operator (action-taker) can interact with the nuclear plant model and only the senior reactor operator (decision-maker) can follow written procedures. Most crew information exchange is related to the execution of proceduralized activities directed by the decision-maker. Information exchanges related to the execution of control panel activities requested by the decision-maker are categorized as ordered activities.

Following information sorting, limited pre-processing is performed. Since internal information resides within an operator's memory it is immediately perceived and acted upon. Consequently, when a communication packet related to the selection of a new goal or strategy, the operator immediately updates his or her memory with the new information. Similarly, new mental beliefs are immediately activated. If a newly activated mental belief is associated with a skill- or rule-based mental procedure, the mental procedure is added to an execution queue for later action. Information related to procedure execution immediately updates the operator's internal procedure place-keeping module to note the new active procedure step. Because both activation of procedures and mental beliefs can be used to activate new mental beliefs, new data related to these information categories are used to update the confidence levels of the operator's mental beliefs (blocks 15 and 27 in Figure 10). Externally derived information from the control panel is not immediately acted upon; instead it is used to update mental belief information and sent to memory for later use during the decision-making module. Externally generated ordered activity information (including manipulation of controls or the verification of plant data) requested by the decision-maker are entered into an action queue when perceived by the action-taker. During the later decision-making process, the action-taker checks the ordered action queue and executes queued actions. Following the information pre-processing stage, the operator will initiate the decision making process.

4.1.2 Control Panel Scanning

One form of self-directed information gathering routinely utilized by control room operators is control panel scanning [75]. An important feature of control panel scanning is that operators will often monitor a subset of parameters more closely based on the ongoing plant status. Within the ADS-IDAC model, this focusing process is controlled by the operator's control panel "scan queue". The scan queue contains a listing of parameters that the operator monitors on a frequent basis. Scan queue parameters may include instruments, alarms, and component states. The number of items contained in the scan queue is limited by the individual capabilities of the operator, the amount of attention the operator can apply to information gathering, and the operator's perception of the current plant state. As the number of monitored items in the scan queue increases, the operator improves his or her ability to accurately assess and diagnosis the plant state.

Two main factors determine which items are included in the operator's scan queue: (1) the maximum size limit of the queue, and (2) the priority level of each item in the queue. The maximum size of the scan queue ($N_{Scan\ Queue}$) is determined by Equation 1.

$$N_{Scan\,Queue} = N_{Baseline}\left(1-\gamma_1\, PIF_{Info\,Load}\right)\left(1-\gamma_2\, PIF_{System\,Criticality}\right) \quad \text{(Equation 1)}$$

The constants $N_{Baseline}$, γ_1, and γ_2 are set in each operator's profile and serve to calibrate the model to the desired operator performance level. $N_{Baseline}$ establishes the maximum amount of information that can be contained in the scan queue while the γ factors ($0 < \gamma_i < 0.1$) set the sensitivity of the dynamic scan queue limit to the information load and system criticality PIFs. Qualitatively, as the information load increases (as indicated by a high value of $PIF_{Info\ Load}$), the scan queue size will decrease to prevent an information overload. If there is a significant degradation in the plant level of safety (as indicated by a high value of $PIF_{System\ Criticality}$), the size of the scan queue decreases to force the operator to focus limited attention resources on the most serious problems.

As an accident scenario progresses, certain events (such as the actuation of an unexpected alarm) will prompt the operator to add new items to the scan queue. When items are added to the scan queue, each item is assigned an initial priority level based on its relative importance. At fixed time intervals, the priority level of each scanned item is reduced if the safety functions associated with the monitored item are not perceived as relevant to the current plant state. The component functional decomposition included in each operator's knowledge base serves as the linkage between functional role of each item in the scan queue and the plant state diagnosis. For example, if the operator determines that a reactor coolant system leak has occurred, pressurizer level will be perceived as a relevant item to the plant state but main steam flow might be considered irrelevant. If the size of the scan queue exceeds the maximum size limit ($N_{ScanQueue}$), low priority items are eliminated until the scan queue size restored to less than the maximum limit. Because items that are considered to be functionally relevant to the current plant state will maintain a high priority level, they will be retained in the queue. In this manner, the items contained in the scan queue are dynamically adjusted to permit the operator to shift focus to areas of the plant that are perceived to be most important while keeping the overall information loading at an acceptable level.

Figure 11 provides a simplified overview of the how the control panel scan queue may evolve over time as new information becomes available to the operator. During normal operation, the operator will generally focus on a high level set of parameters in order to ensure that plant systems are operating normally. In this example, typical parameters may include reactor power output, coolant temperature and pressure, and key water levels. If new information is received, such as actuation of an alarm, the operator may shift attention to parameters closely associated with the alarming condition. Thus, if a low pressurizer water level alarm is noted, the operator may begin to regularly monitor pressurizer water level and the reactor coolant level control system. If adding these parameters to the scan queue results in too much of a monitoring burden, the operator may stop monitoring parameters that are considered to be within normally operating limits (in this case steam generator water levels). If the situation continues to deteriorate, the operator's monitoring activities will focus

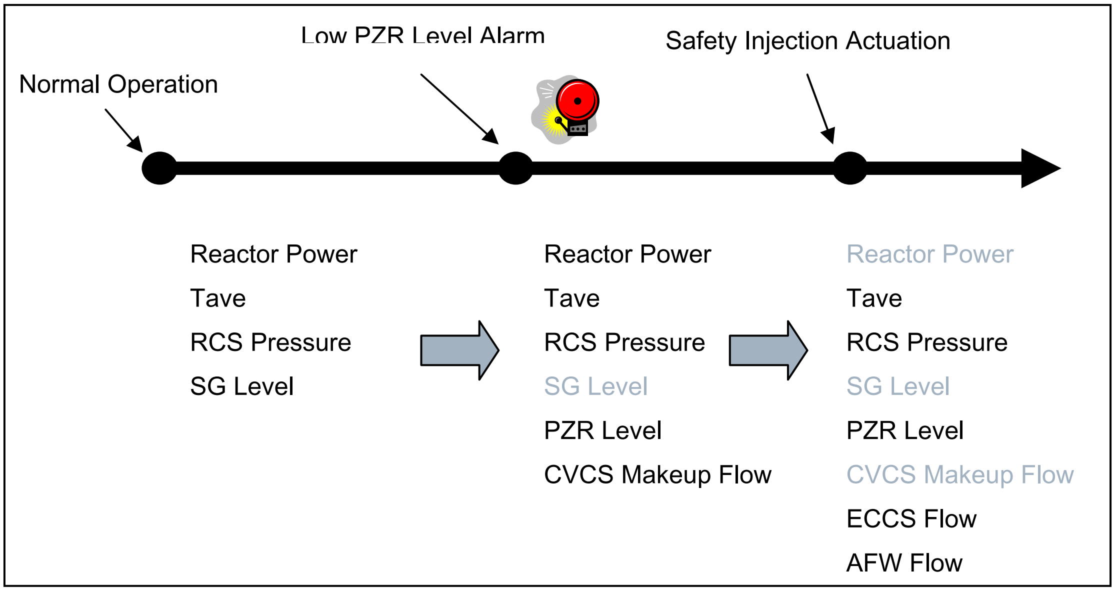


Figure 11 - Dynamic Control Panel Scan Queue

on parameters perceived to be important to the situation, while irrelevant parameters and those that are not problematic may be ignored.

4.2 Information Filtering

As previously noted, the ADS-IDAC operator model does not receive information directly from the plant thermal hydraulic model; instead the information must first be perceived and stored in intermediate memory in order to be utilized for later decision-making activities. The perception process may cause available information to be ignored (either through inattention or deliberate screening) or distorted by biasing. One way information may be missed or otherwise ignored is if a procedure step intended to gather information is skipped through an error of omission. Thus, skipping a step in either a formal written procedure or a memorized procedure can have the effect of censoring information from the operator.

Another way otherwise available information may be withheld from the operator is if the he or she is unable to pay adequate attention to the information due to cognitive limitations. This can occur due to either inattention on the part of the operator or when a period of high information load causes the operator to ignore information that is thought to be less critical. The control panel scan queue provides a means to simulate this type of behavior through adjustment to the maximum scan size permitted for each operator. A higher scan queue limit will allow the operator to monitor more parameters and obtain an improved situational assessment of the plant state. Setting a lower scan queue limit reduces the number of parameters that can be periodically monitored and allows the analyst to simulate the operator's information processing and short term memory limitations. The actual scan queue limit is dynamically adjusted during the simulation and may be less than this input value due to the influence of certain performance influencing factors. When the size of the scan queue exceeds the dynamic limit, low priority parameters are removed form the queue until the size limitation is met.

Finally, ADS-IDAC allows the analyst to establish biasing factors that can be used to either simulate failed instrumentation or influence how the operator perceives information. When no filtering or bias takes place, an operator perceives the actual value of a plant indicator obtained directly from the RELAP thermal-hydraulic model. When a failed indicator or perception distortion is being simulated, a biasing factor is applied to the parameter values obtained from the RELAP model. Several biasing options are available, including additive errors, proportional errors, stuck instrumentation, and instruments that cannot read above or below a preset threshold value. When a parameter is biased in this manner, the operator may use inaccurate data when assessing the procedure step expectations, knowledge-based action prerequisites, and activation criteria for hard wired mental beliefs.

4.3 Operator Knowledge Base

Within the ADS-IDAC environment, cognitive processes for each operator are supported by a unique knowledge base. The knowledge base represents the memorized information, skills, and abilities available to the operator. Specific items included in the knowledge representation include:

- written and memorized procedures,
- diagnostic guidance,
- a functional decomposition model of reactor plant systems, and
- rules governing the activation mental beliefs.

ADS-IDAC allows the analyst to capture both the structure and content of written plant procedures. In general, procedure steps specify an intended action, associated expectations, and contingency actions if the expectations are not met. With this procedure framework, actions, decision points, and procedure transfers can be realistically modeled. The diagnostic guidance portion [16] of the knowledge base supports the inference-based identification of plant events and the selection of knowledge-based actions. The functional decomposition model links reactor plant components to their associated functions, influences the crews' procedural adherence tendencies (i.e., step-skipping), and prioritizes information gathering. Finally, mental beliefs are based on information perceived by the operator and represent discrete observations and decisions. Collectively, mental beliefs characterize each operator's situational assessment of the plant state.

4.3.1 Plant Functional Decomposition

This research effort has extended the capabilities of ADS-IDAC to model the omission of certain procedure steps and the execution of knowledge-based actions. Both of these new features rely on the ability to link needed safety functions to operator's assessment of the current plant state. This linkage is provided by a functional decomposition map that identifies the key plant functions supported by all controls, instruments, and alarms available to the operator. A diagnosis engine supports the operator's plant state assessment by identifying degraded safety functions based on perceived parameter values, component states, and alarms. The relative importance of a procedure step is determined by comparing the functions of the plant equipment referenced by the step to the operator's current plant assessment. Procedure steps associated with equipment that has little relevance to the operator's plant assessment are considered to have a higher likelihood of being skipped. For example, a procedure step associated with activation of a containment pressure suppression system would be more likely to be skipped if the operator had not identified a high containment pressure condition. Additionally, factors such as step complexity, procedure type, and the operator's mental state influence the probability of skipping a procedure step. The operator knowledge base also includes a catalog of actions intended to mitigate degraded plant safety functions. Thus, if the operator diagnoses a safety function failure, an appropriate knowledge-based action can be activated to restore the function.

Step-skipping and the inappropriate execution of knowledge-based actions are two important potential sources of operator errors. Although these behaviors appear to be unrelated, they can both arise from the operator's perception and assessment of the nuclear plant state. For example, a procedure step that requires the operator to perform an action that is thought to be irrelevant to the current plant state (e.g., checking containment pressure when no evidence of coolant leak into containment has been perceived) might be skipped by an operator, particularly when time pressure is high. Similarly, an operator might be tempted to execute a knowledge based action that is thought to directly relate to the perceived plant state (e.g., stopping emergency core cooling injection when the pressurizer water level is believed to be high), even when this action is not directly addressed by the procedures. The common feature shared by both of these examples is the ability of an operator to relate an action to a perceived plant state. Consequently, a central feature of the operator knowledge base is the development of a mental or conceptual model of the reactor plant. A conceptual model allows people to predict the effects of their actions [36]. Therefore, the conceptual plant model in ADS-IDAC serves as the main connection between operator actions and desired consequences. An operator's reactor plant conceptual model is a reflection of his or her education, training, and experience. The ADS-

IDAC functional decomposition provides the framework upon which the operator's conceptual model of the reactor plant is built.

4.3.1.1 Plant Knowledge Structure

In the United States, nuclear power plant operators are licensed by the Nuclear Regulatory Commission and regulated under the provisions of Title 10, Part 55, "Operators' Licenses," of the Code of Federal Regulations (10 CFR 55). In order to obtain a license, an operator must meet certain medical requirements and pass written and operating tests to determine if they can operate the plant competently and safely. Following initial licensing, operators must periodically stand control room watches and pass a requalification program every two years. On the basis of their high level of experience and training, most nuclear plant operators should be categorized as experts in their field.

A number of research studies have identified important differences in the decision-making and problem-solving behavior of experts versus novices. For example, experts develop mental representation of a problem based on underlying structural principles while novices tend to represent problems based on the surface appearance of the problem [19]. Therefore, the method used to categorize plant components in the ADS-IDAC knowledge base should reflect underlying engineering principles rather than physical similarities. While it is possible to categorize plant components based on physical similarities (e.g., categorizing pumps and valves based on component type), a more meaningful structural categorization considers the safety functions supported by the component, linkages to other components, and prerequisite conditions associated with component operation.

Inspired by the Multilevel Flow Modeling technique [18], a functional component categorization based on the flow of energy, mass, and momentum is used in ADS-IDAC. In this modeling scheme, the reactor plant is viewed as a collection of mass, energy, and momentum flow paths, each containing sources and sinks. For example, in a PWR, the reactor core is a source of energy, while each steam generator is considered to be an energy sink. Because the reactor coolant system carries the energy released in the reactor core to the steam generators, any imbalance between energy production and removal will impact the reactor coolant energy state. In general, the following rules are used to identify mass, energy, and momentum imbalances:

- Energy flow imbalances are generally indicated by changes in temperature for subcooled single phase systems and changes in pressure for saturated two phase systems;
- Imbalances between mass sources and sinks are generally related to net inventory measures such as tank or vessel levels; and
- Momentum imbalances are generally indicated by changes in flow rates.

Although this modeling technique provides a powerful mechanism for linking components within a functional framework, a key issue is the level of plant system decomposition used to organize energy, mass, and momentum flow paths. If the decomposition level is set too high, there will be insufficient resolution between component functional groups to differentiate key components from less important ones. If the decomposition level is set too low, the model will not represent the integrated plant functional model typically used by operators.

4.3.1.2 Identification of Key Plant System Groups and Functions

In order to functionally categorize plant components, it is first necessary to identify the flow path boundaries. Plant system groups are used to represent the boundaries for mass, energy, and momentum flow paths. In general, it is desirable to make the plant system group boundaries as broad as possible in order to maximize the ability to link plant components within the operator knowledge base.

The strong coupling among nuclear plant systems presents a significant challenge when identifying functional system groups. Within a nuclear plant, energy flow is often carried by moving fluids such as the reactor coolant or main steam systems; therefore, changes in mass flow rate can directly impact energy flow. Consequently, coupling can result in imbalances in one flow type influencing a second flow type within the same system group or a connected system group. Coupling can also mask the cause of disruption in energy, mass, or momentum flow. For example, changes in reactor coolant system temperature due to an imbalance between reactor core power and turbine load (an energy flow imbalance) can result in variations in system volume due to the expansion or contraction of the coolant (which might be interpreted as a mass flow imbalance). An additional consideration is the diagnostic capability afforded by the system groupings. It is desirable to constrain the system group boundaries such that a flow imbalance within a grouping can be linked to a manageable number of potential causes. In practice, the identification of the system groups requires a balance between maximizing the linkage between plant components, minimizing undesirable coupling, and providing a high level of diagnosticity. Five functional system groups have are used in the current ADS-IDAC PWR model (Table 1).

Table 1 - Pressurized Water Reactor System Groups

System Group	Flow Paths
Reactor Coolant	Energy
	Mass
	Momentum
Pressurizer	Energy
	Mass
Steam Generators(1)	Energy
	Mass
Secondary(2)	Energy
	Mass
Containment	Energy

(1) Each steam generator is considered a separate system group
(2) The secondary system group includes the turbine, main steam, main feed, and condenser systems.

The level of decomposition shown in Table 1 provides sufficient resolution to differentiate between the functions supported by control panel equipment while maintaining the ability to integrate high level plant functions. The operator's assessment of the adequacy within each functional group is based on the perceived trends in energy, momentum, and energy flows. Three trend categories are currently used: stable, increasing, and decreasing. During normal, steady-state operation, the trend for all functional flow paths within each group is stable. Departure from a stable condition indicates that a deficiency in the affected flow path has occurred and mitigative measures are required to stabilize the condition.

4.3.1.3 Component Functional Map

The component map describes the functions associated with every control, indicator, and alarm available to the ADS-IDAC control room crew. Each operator knowledge base includes a unique component functional map in order to match operator behavior to a desired level of knowledge, skills, and abilities. A three parameter coding scheme is used to identify component functions. The first parameter identifies the type of flow (i.e., energy, mass, or momentum). The second parameter identifies the system group that transports the energy, mass, or momentum flow. The third parameter identifies how the component affects (or is associated with) the flow balance in the system group. Thus, a possible component functional code might read: “energy flow, reactor coolant system, energy source”. More than one functional code can be assigned for a single component.

As an example of the functional coding method, consider the functional decomposition of the following three components: (1) the turbine trip alarm, (2) the reactor coolant system loop average temperature, and (3) the manual reactor scram switch. Each of these components is associated with the flow of energy within the reactor coolant system. Specifically, the turbine trip alarm indicates the possible loss of an energy sink from the reactor coolant system, the manual reactor scram switch can be used to reduce a significant energy source to the RCS, and a change in loop average temperature indicates an imbalance between energy sources and sinks. The component functional map allows each of these components to be meaningfully linked within the operator knowledge base. A sample functional decomposition map for a three loop pressurized water reactor is provided in Appendix C.

4.3.2 Mental Beliefs

Mental beliefs represent discrete decisions or observations and serve as the basic decision-making building blocks in ADS-IDAC. Examples of mental beliefs can include basic observations such as "Decreasing Pressurizer Level" or more complicated diagnostic conclusions such as "Possible Steam Generator Tube Rupture". Each mental belief includes profiling parameters that specify when the belief can be activated and the operator actions taken as a consequence of the belief. Rich mental models of plant behavior and complex operator actions can be created by appropriately combining mental beliefs.

The implementation of the mental belief model in ADS-IDAC was inspired, in part, by the Recognition Primed Decision (RPD) model [13]. The RPD model describes decisions made by experienced persons operating under dynamic conditions with time pressure and ambiguous information. Key observations supporting the RPD model are that experienced decision-makers often use a pattern matching process when selecting an appropriate course of action and that the decision process does not usually involve simultaneous evaluation of multiple action alternatives. Pattern matching allows the decision-maker to relate his or her current situational assessment to a memorized prototypical situation. Because prototypical situations are directly linked to typical courses of action, once the decision-maker identifies a sufficient match, an appropriate course of action is apparent. Consequently, the RPD model shifts the decision-making focus from the evaluation of multiple possible alternatives to information gathering and situational assessment. Consistent with the RPD model, ADS-IDAC mental beliefs are triggered by a pattern matching process that compares the operator's perceived situational assessment to a set of prerequisites that define the activation conditions for the mental belief. An additional motivating factor for the implementation of mental beliefs is the observation that nuclear plant operators engage in important situational assessment and decision-making activities in parallel with written procedure following [25]. Mental beliefs provide a means to overlay decision-making tasks with procedure following in order to obtain more realistic operator behavior. For example, mental beliefs can be used to activate actions that are either not adequately described by plant procedures or may be performed in addition to procedural actions. Additionally, mental beliefs can be used to model continuous activities performed in parallel with written procedures.

To demonstrate the use of mental beliefs, an application example involving the control of auxiliary feedwater flow following a reactor a shutdown has been developed. Following the shutdown of a pressurized water reactor, makeup water to the steam generators (SGs) is provided by the auxiliary feedwater (AFW) system. The AFW system generally consists of two main subsystems: one subsystem with motor driven AFW pumps and a redundant subsystem with a steam turbine driven pump. Each SG normally is equipped with two AFW supply throttle valves – one for the motor driven subsystem and one for the turbine driven subsystem. When the AFW is in standby, all throttle valves are normally fully open to provide the maximum available flow to the SGs. Once SG water levels are restored to the desired control band, the operators will adjust AFW flow to match steam demand and minimize reactor coolant system cool down. Although AFW control is an important task, it is often considered to be within the “skill-of-the-craft” for a reactor operator. Consequently, plant procedures may require that the operators adjust AFW flow to maintain SG levels within a specified range, but do not provide detailed guidance on controlling AFW flow rate (e.g., when to open or shut the AFW throttle valves, the magnitude of control adjustments, and the frequency of flow adjustments). The main mental tasks needed to control AFW include: (1) detection of a condition requiring AFW, (2) verification of correct system alignment, (3) detection of a low SG level condition requiring an increase in AFW flow, and (4) detection of a high SG level condition requiring a decrease in AFW flow. The ADS-IDAC mental belief network used to accomplish these tasks is shown in Figure 12.

The AFW control process is initiated when the operator perceives the actuation of the motor driven auxiliary feedwater pump (MDAFP) autostart alarm. Actuation of this alarm activates the “Possible Safety System Actuation” mental belief which will direct the operator to check the operating status of the MDAFPs. If the operator perceives that one of the MDAFPs is running, the “MDAFP Running” mental belief will be activated and the operator will be prompted to verify the AFW system alignment. Following activation of the "MDAFP Running" mental belief, if the operator perceives a low SG level condition (and the MDAFP throttle valve is not

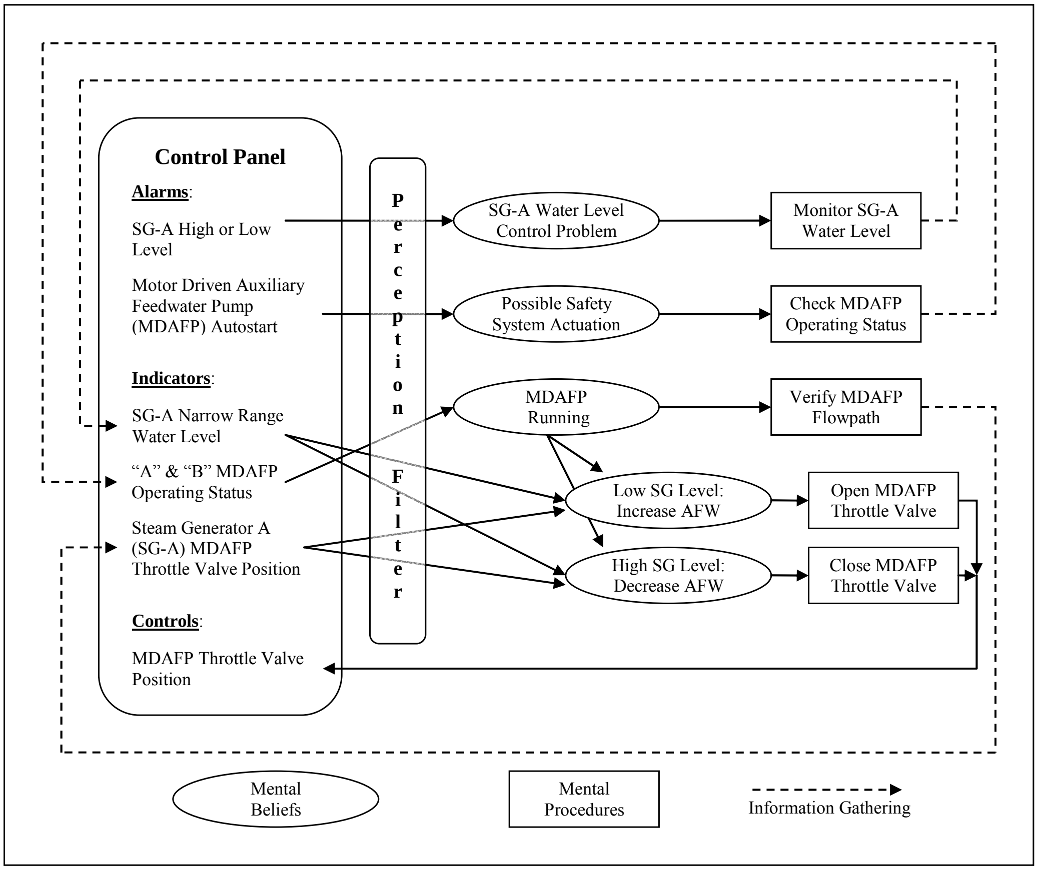


Figure 12 - Task Network for Controlling Auxiliary Feedwater Flow

already fully open), the operator will execute a mental procedure to incrementally open the throttle valve. Similarly, if the operator perceives a high SG level condition (and the MDAFP throttle valve is not fully closed), the operator will execute a mental procedure to incrementally close the throttle valve. An important feature of ADS-IDAC is that the operator only acts on perceived information, rather than the raw data from the thermal-hydraulic plant model. Perception filters can block or distort plant information possibly leading to an inaccurate situational assessment and inappropriate activation of mental beliefs. By adjusting the valve opening/closing increment and the mental belief activation and reset times, the analyst can mimic a variety of AFW control styles (e.g., frequent fine adjustments or infrequent course adjustments).

4.3.3 Procedures

Two main types of procedures are used in ADS-IDAC: (1) written procedures, and (2) memorized mental procedures. Written procedures represent formal proceduralized guidance contained in normal, abnormal, and emergency operating procedures. Memorized mental procedures represent the skill- and rule-based actions routinely used by the operators that do not require formal procedure guidance.

Four general types of procedural actions can be executed: (1) changing the component operating mode (e.g., automatic vs. manual mode), (2) setting a specific control value for a component (e.g., throttling control valve to 50% open), (3) incrementing the control setting of a component (e.g., throttling open a control valve by an additional 10%), and (4) setting a control value based on a perceived parameter (e.g., setting the steam dump target pressure equal to the perceived main steam header pressure). These capabilities provide sufficient flexibility to realistically model all significant operator interactions with the plant model.

ADS-IDAC includes the capability to represent both the structure and content of many types of plant procedures. Procedure step execution follows a standard action-expectation-mitigation format. In this procedure framework, each step specifies the following:

- an operator action;
- a set of expectations that are anticipated to occur as a result of the action; and
- a mitigative action if the expectations are not met.

Within ADS-IDAC, the analyst specifies the content and logic of each step action and several parameters that control how the step is executed. Figure 13 provides an example of the coding of an emergency operating procedure step in ADS-IDAC.

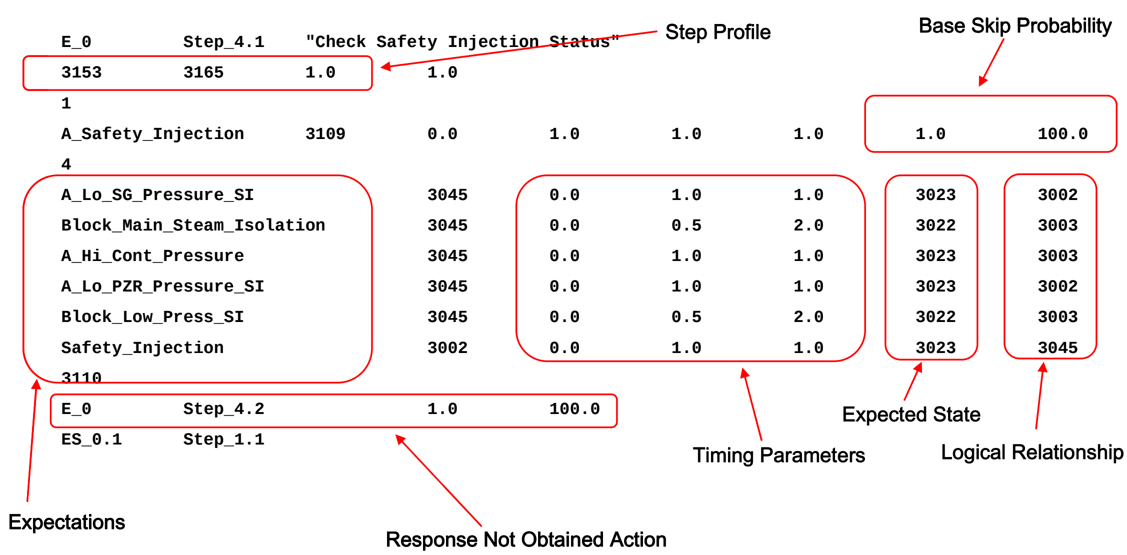


Figure 13 - Example ADS-IDAC Procedure Step

The example instructions shown in Figure 13 are needed to execute an early step in the entry emergency operating procedure for a typical pressurized water reactor. The specific format and content for coding procedure steps in ADS-IDAC is provided in Appendix K. The objective of this step is to check the status of the emergency core cooling system. If emergency core cooling is needed, the operator continues with the procedure; otherwise, the operator transfers to a different procedure to ensure an orderly normal shutdown of the reactor plant. Several key features of the procedure step profile are highlighted:

- Step Profile – The step profile provides a description of the type of procedure (an emergency operating procedure, is coded as "3153" in the ADS-IDAC simulation computer code), the type of procedure step (a diagnosis step, coded as "3165"), and a parameter that indicates the complexity of the step ("1.0").
- Base Skip Probability – The base skip probability is modeled with a Beta distribution and establishes the baseline probability for omitting the step (i.e., an error of omission). In this case, a Beta distribution with $\alpha = 1.0$ and $\beta = 100.0$ is specified (resulting in a mean skip probability of approximately 0.01).
- Expectations, Expected State, and Logical Relationship – In this case four separate expectations are specified:

    i. Existence of conditions that would actuate the low main steam pressure engineered safety feature when this feature has not been previously blocked by the operator;
    ii. High containment pressure;
    iii. Existence of conditions that would actuate the low reactor coolant system pressure safety feature when this feature has not been previously blocked by the operator; or
    iv. Actuation of the emergency core cooling system.

    The logical relationship parameter permits complex combinations of plant states or parameters to be specified. If any of the above expectations are not met, the operator will perform the response not obtained action.
- Timing Parameters – The time required to perform each action or verify the status of each expectation is specified by the analyst using a three parameter Weibull distribution. The Weibull probability distribution is used to capture stochastic variability in crew response time.

- Response Not Obtained Action – If any procedure step expectations are not met, the operator will perform this action. In this case, a procedure transfer to continue the current procedure at step 4.2 is specified. If the expectations are met (i.e., emergency core cooling is not required), the operator will transfer to procedure ES 0.1, Step 1.1.

Generally, a written procedure is continued until the procedure is completed. However, the procedure flow may be interrupted by procedure transfers (which direct the crew to a different procedure), activation of an instinctive response action, or abandonment of the "Follow Written Procedure" strategy. Two types of procedure transfers can be modeled: (1) a permanent procedure transfer and (2) a temporary transfer to an auxiliary procedure followed by resumption of the initial procedure. When a permanent transfer is executed, the original procedure is terminated and will not be reactivated when the new procedure is completed. When a temporary transfer is executed, the original procedure will be recommenced at the step where it was interrupted when the new procedure is completed. ADS-IDAC executes a temporary procedure transfer when the new procedure is either a mental procedure or a functional recovery guideline. An example of the first type of procedure transfer is the transfer from a general reactor trip procedure to a more specific emergency procedure (e.g., transfer from the E-0 to E-3 procedures during a steam generator tube rupture event). The second type of transfer supports implementation of functional recovery guidelines that are used to temporarily interrupt the current procedure to address a degraded condition. A temporary procedure transfer might occur if the operators detect a loss of heat sink following a reactor trip – in this case the operators leave the reactor trip procedure, perform appropriate steps of the loss of heat sink functional recovery guideline, and then return to the reactor trip procedure following restoration of the heat sink function. During a temporary procedure transfer, the operator "remembers" the previous procedure step in effect before the transition and returns to this step after the temporary transition procedure is completed. Because only the Decision Maker is permitted to direct the performance of a written procedure, ADS-IDAC places restrictions on the types of procedure transitions available to each operator. The Action Taker may initiate a memorized mental procedure and transition to other mental procedures, but may not initiate or transition to a written procedure. The Decision Maker may initiate and transition between all procedure types.

Four types of event sequence branches can be generated during procedure execution: (1) mental procedure activation time branches, (2) action execution time branches, (3) action control value branches, and (4) step-skipping. After a mental belief is activated, the associated memorized mental procedure is initiated after the activation time delay has elapsed. Mental procedure activation time branches allow the analyst to examine the impact of variations in the time delay between the perception of conditions that activate the mental belief and the execution of the associated skill- or rule-based actions. Action execution time branches enable multiple event sequence branches to be generated to model variations in the time taken by the control room crew in performing procedure actions. Action control value branches can be used to model variations in control inputs such as control valve positioning and the setting of control system target setpoints. Finally, procedure step-skipping branches model the omissions of procedure actions based on the relevance of the step actions to the operator's situational assessment. The ADS-IDAC step-skipping model is described in more detail in Section 6.2.

5. Decision-Making Module

The decision-making process immediately follows the information processing stage. During the decision-making process each operator assesses current plant conditions, evaluates his or her current high-level goal and associated strategy for achieving the goal in light of his or her situational assessment, identifies specific actions in accordance with the selected problem-solving strategy, and implements memorized skill- and rule-based activities through the activation of mental beliefs.

5.1 Decision-Making Within ADS-IDAC

The decision-making process is the heart of the ADS-IDAC approach to crew modeling. All crew interactions with the nuclear plant model are identified through the decision making process. Because the decision-making engine is based on an information driven architecture, small variations in perceived information may lead significant changes in the output from the decision-making process. For example, biasing or ignoring a critical parameter or other piece of information may lead the crew to initiate inappropriate actions in response to an accident. It is this feature that enables ADS-IDAC to show promise for predicting and analyzing errors of commission.

The decision-making process is shown in Figure 14. The decision-making

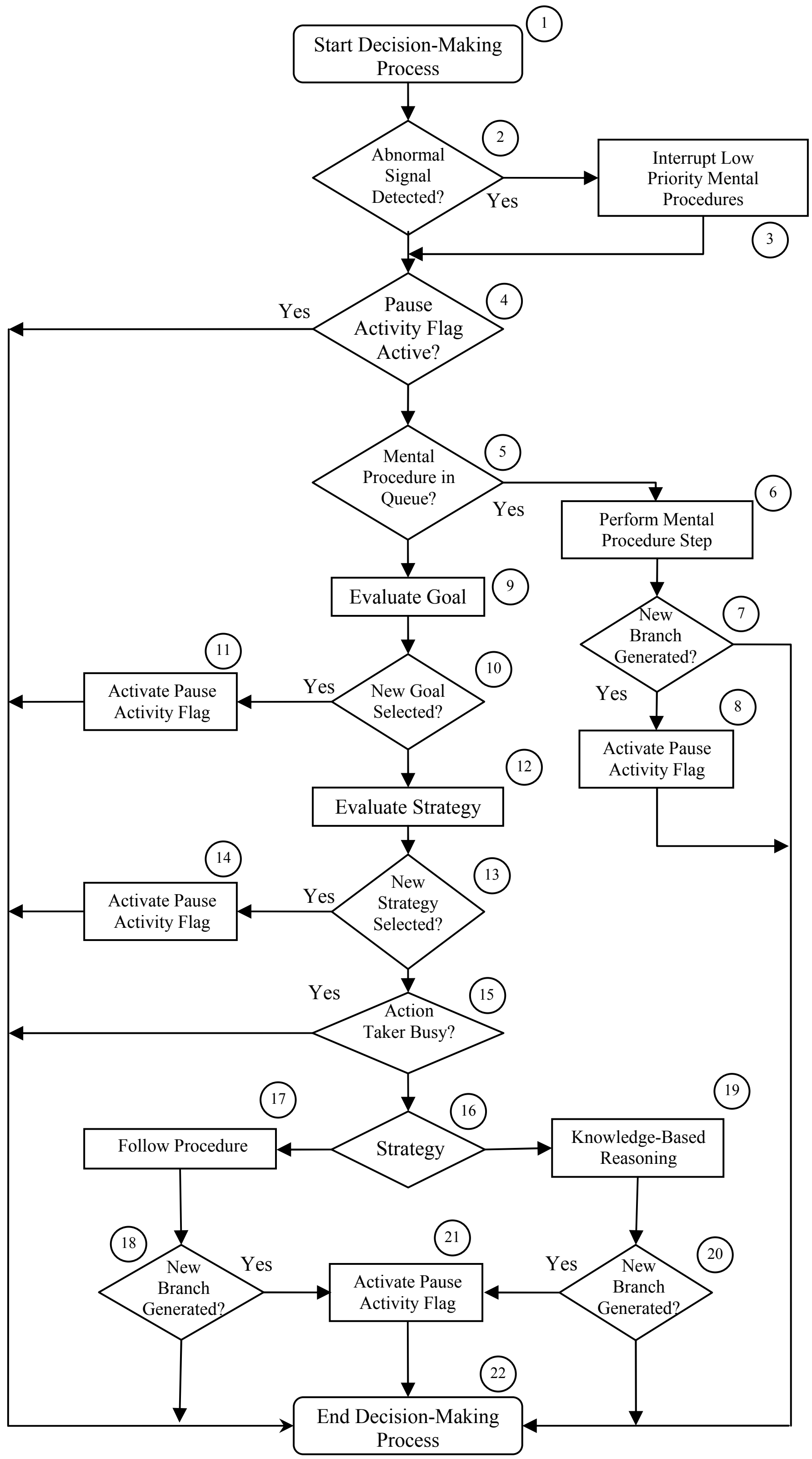


Figure 14 - Decision-Making Process (Decision Maker)

begins with an update to the operator's situational assessment based on new information perceived during the information processing stage (Block 1). The diagnostic process updates the operator's assessment of the status of key plant functions (see Section 4.3.1.2) and the status of the various accident categories identified in the operator knowledge base (described in greater detail in Section 5.5.1.2). After the operator's situational assessment is updated, a determination is made if an abnormal signal has been detected (Block 2). In this context, an abnormal signal refers to a condition that is incompatible with continued power operation and requires the operators to shift their focus to an accident response mode. An abnormal signal is defined as the perception of a reactor trip condition[5] or the relationship value[6] in any accident related event diagnosis exceeding a pre-established threshold. The threshold value used for this determination is one of the static PIF factors included in the operator profile (see Section 7.2.2). If an abnormal signal is detected, the operator will suspend implementation of any in-progress low priority activities that may have been previously initiated. In some cases, these low priority activities have limited mitigative impact during an accident or are incompatible with plant conditions (e.g., reducing turbine load is no longer needed once the reactor has tripped).

The next phase in the decision-making process is to determine if any skill- or rule-based were previously activated and added to an action queue (Block 5). If any

[5] The operator knowledge base includes two special reserved mental beliefs: "Reactor_Tripped" and "Normal_Operation". The "Reactor_Tripped" mental belief is used to diagnosis an abnormal condition, while the "Normal_Operation" belief is used for the goal selection process. The conditions that activate these beliefs are not pre-defined and the analyst can assign appropriate prerequisite conditions to each these mental beliefs.
[6] For the purposes of this discussion, the term relationship value can be loosely interpreted as the operators' confidence level in the associated event diagnosis.

such actions exist, the action is selected for execution during the Action stage and further decision-making activities are bypassed. This process prioritizes the execution skill- and rule-based actions over procedurally driven activities.

The operator next evaluates the appropriateness of the current goal (Block 9) and strategy (Block 13). If the goal or strategy needs to be updated, further decision-making activities are suspended and the new goal or strategy is sent to the communication blackboard for implementation during the next information processing phase. If no goal or strategy updates are needed, the decision-maker verifies the availability of the action-taker to execute ordered activities (Block 15). Since the decision-maker cannot operate control panel equipment, the action-taker must execute any action that the decision-maker needs to perform that requires the control panel. Therefore, if the action-taker is not available to receive and execute an ordered action, further decision-making is suspended until the action-taker is available. If the action-taker is able to receive an order from the decision-maker, an appropriate problem-solving approach is followed based on the selected strategy (Block 16). In the current implementation of ADS-IDAC, the procedure-following and knowledge based reasoning strategies cannot be active simultaneously. Once an appropriate action is identified in a procedure (Block 17) or using a knowledge-based approach (Block 19), the decision-making process ends. The identified action is executed during the Action execution phase.

5.2 Goal Selection

The selection and verification of a high-level goal is the one of the first steps in the decision-making process. ADS-IDAC currently supports four high level goals:

- maintain normal operation;
- monitor;
- troubleshoot abnormal conditions; and
- maintain global safety.

Each of these goals drives specific operator behaviors. For example, selection of the "troubleshoot abnormal conditions" goal enables the activation of knowledge-based actions. Activation of the "mitigate accident conditions" goal initiates the emergency operating procedures if procedure usage is enabled for the operator. The selection of a specific goal is based only on information perceived by the operator and the status of the static performance influencing factors associated with goal selection (see Section 7.2). In general, the decision-maker selects the high level goal for the operating crew; the other operators will then update their respective goals to match the decision-maker's goal.

The decision-maker goal selection process is shown in Figure 15. As discussed in Section 5.1, the operator re-evaluates their high-level goal every time the decision-making process is activated. The first step in the goal selection process is an assessment of the current plant conditions (Block 2). The operator knowledge base includes a "Normal_Operation" mental belief that describes of the plant parameters associated with a normal full power operating condition. Although this description can be customized to reflect variabilities in operator knowledge, a typical normal operating condition description for a pressurized water reactor might include the following elements:

- Reactor Coolant System Temperature
- Reactor Coolant System Pressure
- Pressurizer Level
- Steam Generator Water Levels

The analyst should specify the expected values for these parameters during normal plant operating conditions. Additionally, the analyst can adjust a threshold parameter

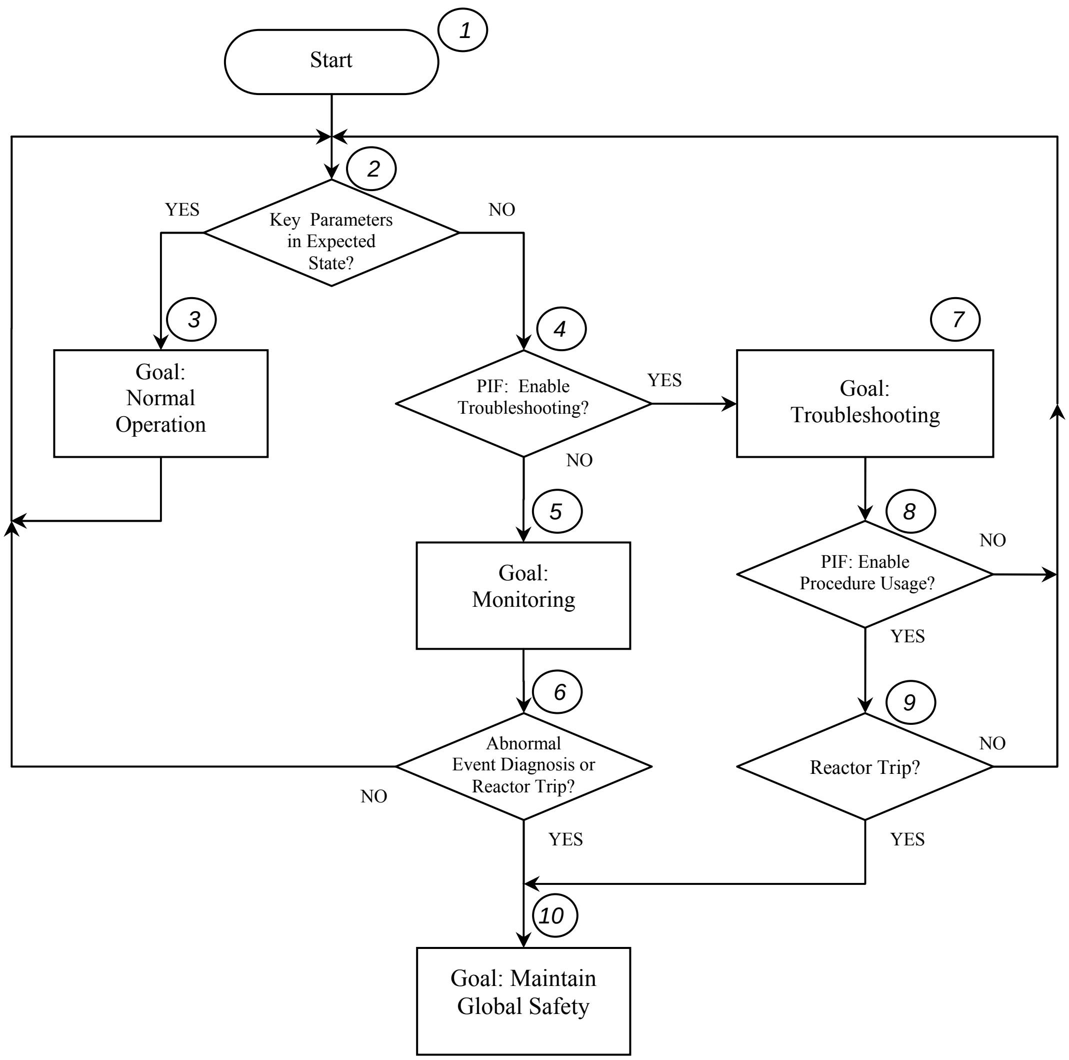


Figure 15 - Goal Selection Process (Decision Maker)

that establishes how many of the specified conditions must be met in order for the operator to conclude that the plant is in a normal operating state. If a the "Normal_Operation" mental belief is activated, the operator selects a goal of maintaining normal operation. If the plant is not normally operating, the operator selects either the a monitoring or troubleshooting goal depending on the preferences specified in the operator profile (Block 4, see Section 7.2.3). If a monitoring goal is selected, the operator will continue information gathering activities to improve the assessment of plant status. If the operator detects an abnormal plant condition or determines that a reactor trip has occurred (Block 6), the maintain global safety goal is activated (the criteria for assessing Block 6 is identical to the criteria used in Figure 14, Block 2). If a troubleshooting goal is selected, the operator will initiate knowledge-based actions based on identified plant functional needs. The operator can transition from troubleshooting to maintaining global safety margin only if a reactor trip condition is detected. The operator may return to a maintain normal operation goal from either the monitoring or troubleshooting goal if plant conditions return to a normal status. However, once the maintain global safety goal is activated, the operator cannot transition to a new goal (i.e., the operator may not return to either the normal operation, monitoring, or troubleshooting goals). Following evaluation and selection of an appropriate goal, the decision-making process proceeds to problem-solving strategy selection.

5.3 Problem Solving Strategy Selection

The problem solving strategy establishes the overall approach the crew uses to achieve their selected goal. Four problem solving strategies can be used to achieve high level goals:

- Wait and Monitor – a passive information gathering strategy intended to improve the operator's situational assessment;
- Instinctive Response – perform simple skill- or rule-based actions that are activated by matching perceived information to memorized situation-response profiles;
- Follow Written Procedures – implement formal written procedures (e.g., abnormal or emergency operating procedures); and
- Knowledge-Based Reasoning – use a diagnostic process to guide crew actions in order to balance the flow of mass, momentum and energy within plant systems.

The strategy selection process for the decision-maker is shown in Figure 16. To ensure that crew actions are coordinated, an order of precedence for problem solving strategies has been developed. The following rules guide the transition between operator problem solving strategies:

- The "Wait and Monitor" strategy has the lowest order of precedence and is only activated if no other problem solving strategy is active;
- The "Knowledge-Based Reasoning" and the "Follow Written Procedure" strategies are mutually exclusive and cannot be activated simultaneously;

- The implementation of high priority "Instinctive Response" strategy actions will interrupt all other strategies. Lower priority instinctive response actions may interrupt other strategies depending on the crew's high level goal and the individual operator profile. Once the instinctive response actions are complete, the operator will return to the previous strategy.

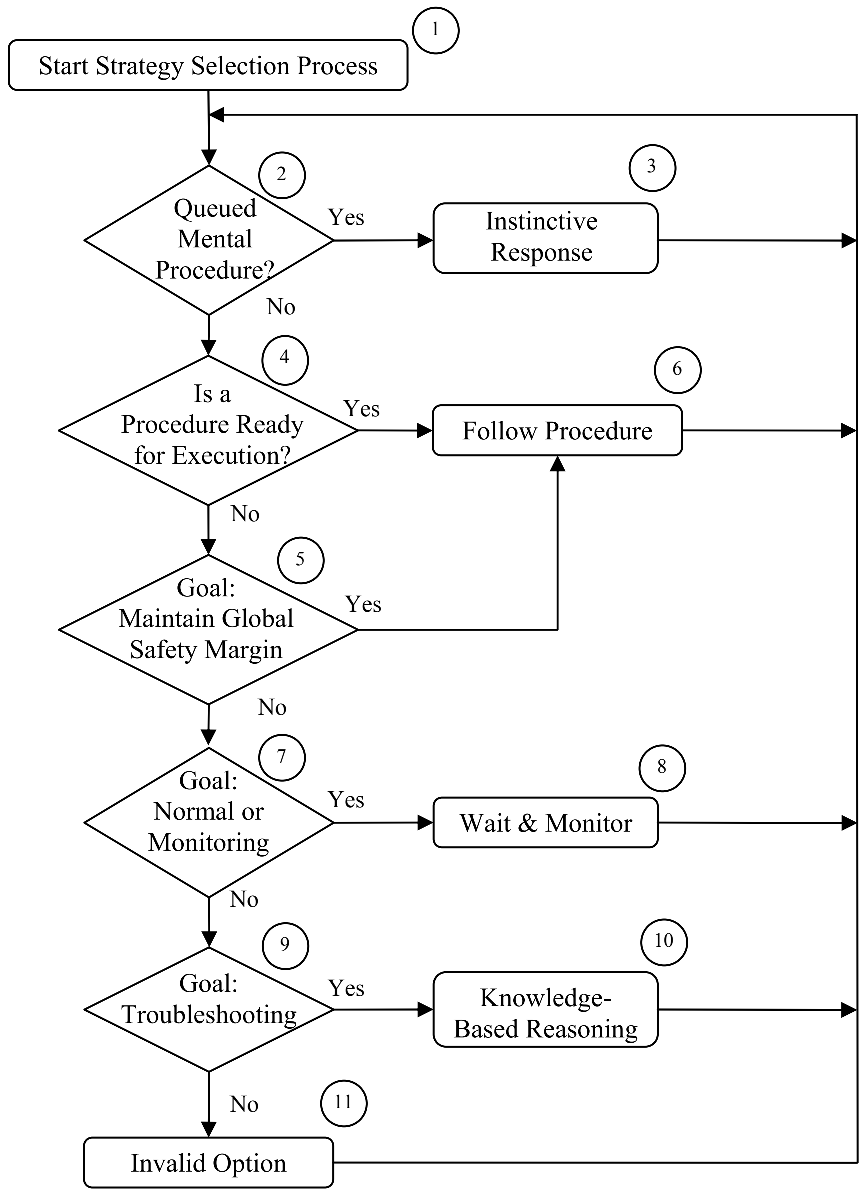


Figure 16 - Strategy Selection Process (Decision Maker)

The main determinant in selecting a strategy is the operator's high level goal. However, because a memorized mental procedure can initiate the emergency operating procedures, if the decision-maker has a queued written procedure for execution (Block 4), the procedure following strategy can be activated regardless of the high-level goal. For example, a mental memorized procedure might cause the operator to initiate the emergency procedures in order to execute a controlled shutdown of the reactor plant (thus queuing the appropriate written procedure). This strategy transition criterion will enable the operator to initiate the emergency procedures prior to selection of the "Maintain Global Safety Margin" goal.

5.4 Mental Beliefs

Within ADS-IDAC, mental beliefs represent discrete decisions or conclusions that the operators reach based on their situational assessment. Three main types of mental beliefs are used in ADS-IDAC: (1) symptom-related mental beliefs that support the event diagnosis process, (2) rule-based mental beliefs that trigger memorized procedures, and (3) mental beliefs that represent intermediate decisions and are used as prerequisites for building more complex beliefs. An example of how a network of mental beliefs can be used to model complex operator behaviors is described in Section 4.3.2.

Mental beliefs include a number of properties that describe the conditions required to activate the belief and resultant actions. Each mental belief is associated with prerequisite conditions that describe the prototypical situation that activates the belief. Prerequisites may include alarms, component states, parameter values, active procedures, and other mental beliefs. A confidence level activation threshold is used to specify how many of the specified prerequisite conditions must be met in order to activate the associated mental belief. Three parameter Weibull probability density functions are used to model an activation time delay and a reset time delay for each belief. The activation time delay specifies the time lag between activation of the mental belief and execution of the associated action. The reset time delay is used to control repeated activations of a mental belief. When a mental belief is activated, a reset timer is started and subsequent reactivation of the mental belief is blocked until the reset timer has expired. The reset capability is needed to model skill- or rule-based actions that must be performed in a repetitive manner (such as adjustment of AFW flow following a reactor trip). The activation status of all mental beliefs is updated when the operator perceives new information.

Each mental belief may be associated with a mental procedure. Mental procedures specify memorized skill- and rule-based actions typically performed by an operator without reference to a written procedure. In general, mental procedures cover the following broad categories:

- Alarm response procedures – information gathering intended to confirm the conditions associated with the alarm and improve the operator's situational assessment;
- Diagnostic procedures – directed information gathering in order to identify a specific component problem. Diagnostic procedures are used when the operator needs to use more complicated logical inference than can be accomplished with mental beliefs alone;
- Control procedures – memorized procedures that represent automatic skill-of-the-craft actions performed by control room operators. Adjustment of auxiliary feedwater flow following a reactor shutdown is an example of a typical control procedure;
- Mitigation procedures – memorized procedures intended to mitigate degrading plant conditions (e.g., manual safety injection actuation following perception of low pressurizer level).

The content of mental procedures is guided by two fundamental principles: (1) the actions can be performed without reference to a written procedure and (2) the procedure can be accomplished within a short time period (generally within a few minutes). The first principle limits mental procedures to relatively simple skill- or rule-based tasks. The second principle prevents a single task from monopolizing the operator's attention during rapidly changing events. If needed, complex or lengthy tasks can be decomposed into smaller discrete tasks to meet these guiding principles. Upon activation of a mental belief, the associated mental procedure (if one is specified) is added to a queue list in the operator's memory. The order that queued mental procedures are executed depends on the specified priority level for the procedure (higher priority procedures are performed first) and the activation time delay. A sample of typical mental beliefs is provided in Appendix D. A detailed discussion of this topic is also included in Appendix K.

5.5 Diagnosis and Situational Awareness

In general, previous research efforts in nuclear plant accident diagnosis methods have focused on the development of control room operator aids to improve operator diagnostic capabilities during plant events (e.g. [76-78]). However, the focus of this research was the development of a model capable of approximating an operator's heuristic approach to event diagnosis. Consequently, the approach attempted only to obtain reasonable results, rather than the best or most accurate plant diagnosis for a given set of conditions. Based on a review of the available literature on nuclear plant diagnosis methods, it was determined that a fuzzy-logic inference method provided the best diagnostic approach for the ADS-IDAC application. In particular, the fuzzy-logic method is capable of representing a large amount of operator knowledge in the relatively compact form of a symptom-event relationship table. The method also accommodates probabilistic uncertainty in the diagnosis process and avoids the need to develop prescriptive diagnostic logic rules.

5.5.1 Diagnosis Model Description

An event-symptom matrix was constructed to specify the probabilistic relationship between a set of plant symptoms and events. The probability values in the matrix can be interpreted as the probability of observing a particular symptom given that a specific event has occurred. During the ADS-IDAC simulation, the operator's confidence level for each plant symptom is periodically recalculated based on data obtained from the plant thermal-hydraulic model and the operator's ability to perceive this data. Standard fuzzy-logic mathematical techniques have been used to evaluate the likelihood of plant events given a set of input symptoms [79-81]. These techniques provide a lower and upper probability bound for each event. While both the upper and lower bounds are related to the confidence level in perceived symptoms, the lower bound indicates the degree to which all the symptoms associated with a given event have been observed. The upper bound estimate indicates if the symptom confidence levels are consistent with the expected values for the event.

A limitation of the fuzzy inference method is its inability to discriminate among incompatible events when provided with contradictory symptom information. Additionally, a likely event identified by the method may only explain a subset of the observed symptoms. Consequently, a combination of two or more events may be needed to account for the full spectrum of observed symptoms. Although these limitations reduce the usefulness of this approach as a diagnostic operator aid, they provide an advantage within the ADS-IDAC environment since a larger spectrum of possible misdiagnoses may be examined.

5.5.1.1 Symptom Selection

The selection of symptom inputs was guided by several goals: (1) provide a realistic representation of the plant information readily available to the control room operators, (2) ensure the number of symptoms was consistent with the limitations of the operator's short term memory; and (3) adequately discriminate among similar initiating events. Consequently, ten symptoms were selected for this model:

- Reactor Coolant System (RCS) Average Temperature
- Reactor Power
- RCS Pressure
- RCS Loop Flowrate
- Pressurizer (PZR) Water Level
- Containment Pressure
- Steam Generator (SG) Water Level
- Feed Water (FW) Flowrate
- Main Steam (MS) Flowrate
- SG Pressure

With the exception of containment pressure, PWR plant control room operators typically monitor these indicators on a continuous basis. Although not continuously monitored, a containment pressure indicator is available to the operators and was included to provide discrimination between coolant leakage events occurring inside and outside containment. Three possible states were considered for each symptom: increasing trend, decreasing trend, or steady state. Within the ADS-IDAC simulation environment, an operator's confidence in each of these symptom states is specified by a parameter value between 0 and 1.0.

5.5.1.2 Event Selection

Events were selected and categorized to ensure compatibility with the high-level operator goals used in the ADS-IDAC cognitive model. Examples of high-level goals include "maintain normal operation," "troubleshoot," or "maintain global safety." Within the ADS-IDAC environment, an operator's problem-solving behavior is a product of his or her high-level goals and mental state. The operator's mental state includes factors such as cognitive mode, emotion, stress, and perception. Because an operator's high-level goals and problem-solving strategy can influence the selection of an event diagnosis, events were grouped into four broad categories:

- Normal Operating Events - events that are expected to occur with a relatively high frequency (i.e., at least several times a year) during normal plant operation and would not typically preclude continued power operation. These events include normal power level changes and control system failures.
- Anticipated Operational Occurrences (AOOs) – abnormal events that are expected to occur one or more times during the life of the plant and are not expected to cause fuel damage. AOOs include events such as turbine trips, losses of main feedwater, and losses of reactor coolant flow.
- Accidents – abnormal events that could result in fuel damage and significant radiological consequences. Examples of accidents include significant losses of primary or secondary coolant, SG tube ruptures, and anticipated transients without a reactor scram.
- Imbalances - events that identify mass or energy flow imbalances within nuclear plant subsystems. Use of this event initiating class was motivated, in part, by the

multilevel flow modeling technique [18] which decomposes complex systems into mass and energy flow paths. The imbalance diagnosis category supports ADS-IDAC problem-solving strategies associated with the restoration of an imbalanced parameter back to a balanced condition.

Each of these event categories can be associated with specific high-level operator goals and problem-solving strategies. For example, if a set of perceived symptoms results in the identification of both a normal operating event and an accident event, an operator would be more likely to select the normal operating event if his or her high level goal was to “maintain normal operation." The same set of plant symptoms can result in different diagnoses depending operator’s high level goal and problem-solving strategy.

5.5.1.3 Event-Symptom Matrix

Implementation of the fuzzy logic diagnostic approach requires the creation of a two-dimensional numerical matrix that describes the relationship between symptoms and events. Each numerical value in the matrix represents the strength of the relationship between a symptom and an event. A value of 0.0 indicates that the symptom and event are unrelated, while a larger value (up to a maximum value of 1.0) indicates a stronger relationship. In general, a higher relationship value indicates that operator has a greater level of confidence that the symptom would be observed given that the associated event has occurred. Because the relationship matrix is intended to represent the operator’s mental model of plant behavior, numerical relationship values are assigned using heuristic rules or preferably an expert elicitation process rather than a formal thermal-hydraulic analysis. For the purposes of this research study, the following heuristic rules were used to grouped event symptoms into the broad categories of primary, secondary, and tertiary symptoms:

- Primary symptoms directly relate to the initiating event and are expected to be observed with a high degree of confidence;
- Secondary symptoms are the result of the primary symptoms and are normally expected to be observed, but with a lower degree of confidence than primary symptoms; and
- Tertiary symptoms may arise due to the presence of primary or secondary symptoms but can be mitigated by either control system operation or thermal hydraulic feedback mechanisms. Consequently, tertiary symptoms may not be observed and are assigned a low degree of confidence.

After the symptoms of each event were categorized, relationship values were assigned based on engineering judgment and the guidelines of Table 2.

Table 2- Symptom-Event Relationship Values

Symptom Type	Relationship Value Range(1)
Primary	0.7 – 1.0
Secondary	0.4 – 0.7
Tertiary	0.1 – 0.4

(1) A higher value indicates a stronger relationship between the symptom and event.

It should be noted that the fuzzy logic quantification approach is based on set membership (i.e., determining if a specific event is included within the membership of a set of events associated with the observed symptoms). Therefore, the quantification results are expressed in terms of membership values rather than a true probability or degree of confidence. Membership values range from 0.0 to 1.0, with higher membership values indicating a stronger relationship between the event of interest and the perceived symptoms. It is possible that a thorough and carefully executed expert elicitation process using experience control room operators may be able to strengthen the conclusions that can be drawn from this fuzzy logic quantification approach. However, since this research project was intended to demonstrate only the feasibility and usefulness of this diagnostic approach, use of a controlled expert elicitation process to populate the relationship matrix was deemed to be outside the scope of this research effort. Consequently, the results obtained from the diagnostic approach are expressed in terms of membership or relationship values, rather than true probabilities.

To illustrate the heuristic method for assigning relationship values, one can consider the occurrence of an uncomplicated reactor trip. The primary symptoms for this event include decreasing core power due to insertion of the control rods and a consequent decrease in the RCS average coolant temperature. Decreasing pressurizer water level is considered to be a secondary symptom since it is caused by the contraction of the primary coolant due to the decrease in average coolant temperature. Decreasing reactor pressure is considered to be a tertiary symptom because operation of the reactor pressure control system will tend to mitigate the pressure decrease caused the decreasing pressurizer water. Therefore, decreasing core power and decreasing average temperature would be expected to be the most likely symptoms to be observed and would be assigned a high relationship value. Decreasing pressurizer water level would be assigned a mid-range relationship value, and decreasing reactor pressure would be assigned a low relationship value.

A simplified procedure was used to assign relationship values for symptoms associated with imbalance events. In order to increase the likelihood of obtaining a high level of confidence in an imbalance diagnosis, it is necessary to minimize the number of symptoms associated with each imbalance event. Therefore, nuclear plant was decomposed into several major subsystems in order to associate each imbalance event with a single primary symptom (Table 3).

Table 3 - Primary Indicators for Imbalance Events

System	Imbalance	Primary Indicator
Pressurizer	Energy	Pressurizer Pressure
Reactor Coolant	Mass	PZR Water Level
	Energy	Average Loop Temperature
SG (secondary side)	Mass	SG Water Level
	Energy	SG Pressure

A summary event-symptom relationship table is presented in Appendix A.

The quantitative output from the diagnosis engine is related to the membership relationship of each event to the perceived symptoms. If the set of perceived symptoms matches the symptom set of a certain event, that event will have a high membership value. A consequence of the fuzzy logic approach is that an event may have a high membership value even if it only accounts for a subset of the perceived symptoms. Therefore, the fuzzy inference method is unable to discriminate among events that only partially explain all perceived symptoms or resolve contradictory symptom information. These limitations were partially addressed by multiplying the event membership value by a correction factor based on the fraction of perceived symptoms explained by the event. Because the plant functional decomposition is intended to break complex plant behaviors into smaller functional units, the correction factor is not applied to imbalance events. It is also worth noting that the diagnostic limitations of the fuzzy inference method can be advantageous when attempting to model the confusion or biases that real operators might experience.

5.5.1.4 Diagnosis Quantification

Each event symptom in the ADS-IDAC simulation is represented by a set of plant state expectations that may include alarm status, plant parameter data, and component operational states. For example, the event symptom “Decreasing Pressurizer Water Level” can be represented by the three plant state expectations: (1) a low pressurizer water level alarm, (2) a decreasing trend rate for pressurizer water level, and (3) a pressurizer water level below a threshold value. The operator’s confidence in an event symptom is the ratio of expectations that have met the specified criteria to the total number of expectations specified for the event symptom. Following the calculation of event symptom confidence, the relationship for each event is determined. The event relationship value is represented by two probability values: a lower bound (LB) estimate and an upper bound (UB) estimate. The probability bounds are calculated using Equations 2 and 3:

$$LB_j = Min[S_i], where \{i : r_{ij} > 0\} \qquad (Equation\,2)$$

$$UB_j = Min[1, (1 + S_i - r_{ij})] \qquad (Equation\,3)$$

Where S refers to the symptom confidence level, i is the symptom set index and ranges from 1 to N (where N is the total number of symptoms considered in the model), j refers to the event set index and ranges from 1 to M (where M is the total number of events considered in the model), and r_{ij} refers to the event-symptom matrix value representing the expected probability of observing symptom i given than event j has occurred.

5.5.2 Diagnostic Examples

To illustrate the use of the fuzzy-logic diagnostic inference method, two example cases are presented in Section 5.5.2.1 and 5.5.2.2. The first example demonstrates the capability of the fuzzy-logic diagnosis method to identify a range of possible initiating events given a set of hypothetical symptoms. The second example shows how the diagnosis method can be used to replicate actual nuclear plant misdiagnosis events obtained from industry operating experience. The same event-symptom matrix was used for each of the two examples to demonstrate the flexibility of the method over a wide range of events.

Case 1: Steam Generator Tube Rupture Diagnosis

A steam generator tube rupture (SGTR) is caused by the failure of the SG tube pressure boundary between the reactor coolant and the secondary coolant systems. The pressure boundary failure diverts reactor coolant from the pressurizer to the secondary side of the associated SG. This results in a decreasing pressurizer water level, decreasing RCS pressure, and an increase in SG water level. To compensate for the addition of reactor coolant into the SG secondary, the SG water level control system will tend to decrease feedwater flow. In this basic example, the following symptoms were used as input for the model (Table 4):

Table 4 - SGTR Example Symptoms

Symptoms	Confidence Level
FW Flowrate Decreasing	0.4
PZR Water Level Decreasing	0.5
RCS Pressure Decreasing	0.3
SG Water Level Increasing	0.7

The confidence level can be interpreted as a measure of the degree of belief an operator has about a specific symptom. For example, a confidence level of 0.5 indicates that an operator believes with 50% certainty that the associated symptom represents a true change in plant status rather than a minor transient condition or background noise. Relationship levels of less than 100% could be due to a lack of confirmatory information or ambiguous plant data. During the ADS-IDAC simulation, the operator's confidence level for each plant symptom is periodically updated based on newly perceived information. In this example, application of the diagnostic method resulted in the identification of the following possible events (Table 5). In this case, a diagnosis of an SGTR event has a high probability of occurrence (lower bound of 0.3 and an upper bound of 0.8). The relationship values for events within the imbalance class tend to be higher than events in other event classes since they rely only on a single symptom input. It should be noted that while only the SGTR event can explain all of the observed symptoms, the method does not rule out other possible diagnoses. For example, the observed symptoms could result

Table 5 - SGTR Diagnosis Results

Event Category	Initiating Event	Relationship Value	
		Lower Bound	Upper Bound
Normal Operation	Controller Failure - PZR Water Level	0.3	0.6
	Controller Failure - RCS Pressure	0.3	0.4
	Controller Failure - SG Water Level	0.0	0.4
Accident	Leak – RCS System	0.3	0.7
	Loss of Coolant Accident	0.3	0.4
	SGTR	0.3	0.8
Imbalance	Mass Imbalance – RCS (low mass)	0.5	0.8
	Mass Imbalance - SG (high mass)	0.7	1.0
	Energy Imbalance - PZR (low energy)	0.3	0.6

from either a simultaneous failure of the pressurizer water level, RCS pressure, and SG water level control systems or an RCS leak in combination with a SG level control failure. Identification of a range of possible initiating events allows examination of conditions that may result in operator misdiagnosis. It should also be noted that three functional imbalance events had high relationship values based on the perceived symptoms: low reactor system mass, high SG mass, and low pressurizer energy. All of these functional imbalances are consistent with the diversion of reactor coolant form the pressurizer to the ruptured steam generator.

Within the ADS-IDAC environment, the selection of a likely diagnosis among these alternatives is probabilistically modeled and is influenced by the operator's mental state, high level goals, and problem-solving strategy. Consequently, an operator with a high-level goal of "maintaining normal operation," might select a diagnosis within the normal operation event category such as a control system failure rather than believing that an event within the accident category has occurred. Similarly, an operator using a problem-solving strategy involving the elimination of mass and energy imbalances would be more likely to select a diagnosis within the imbalance event class (e.g., "high SG mass balance") rather than an event within the normal operation or accident event categories. Thus, the same set of symptoms can generate a variety of plausible diagnosis options and follow-up actions.

The ADS-IDAC environment models the operator's follow-up actions in response to an initiating event by simulating operator actions that change the plant state (e.g., initiating emergency core cooling systems, shutting isolation valves, or starting pumps). For example, a diagnosis of an SGTR could be mapped to the actions contained in the appropriate emergency operating procedure. Similarly, a diagnosis of a high mass imbalance in the SG could lead the operator to take actions such as reducing feedwater flow or increasing SG water loss by increasing steam demand. By simulating operator follow-up actions based on a range of possible diagnoses, operator actions that result in further plant state degradation can be identified.

Case 2: Pressurizer Spray Valve Malfunction Diagnosis

During a startup at the Crystal River Unit 3 nuclear plant on December 8, 1991, control room operators were unable to promptly diagnosis a failed open pressurizer spray valve [82, 83]. The operators incorrectly concluded that the cause of an observed reactor pressure decrease was an increase in steam load and withdrew the control rods several times in an attempt to increase the average reactor coolant temperature and reactor pressure. The operators were unable to stop the continuing pressure decrease and the reactor automatically tripped due to low reactor pressure. Following the reactor trip, an operator, believing that the cause of the pressure decrease would be quickly brought under control, bypassed the emergency core cooling system to prevent automatic actuation of the system [84].

During the event, the operators believed that they observed an increase in steam demand, a reduction in average coolant temperature, and an RCS pressure decrease. This perception may have been biased by earlier operations tasks involving changes in steam demand. However, later investigation identified that both steam demand and average coolant temperature were stable during the event. The difference between the actual and perceived symptoms might be explained by a confirmatory bias which led the operators to misinterpret information in order to support their perceived plant status rather than objectively assessing all available information. Consequently, two sets of symptoms were used to characterize the event: (1) the operators' perceived symptoms and (2) the actual event symptoms (Table 6).

Table 6 - Spray Valve Malfunction Symptoms

Symptoms	Relationship Level (Perceived)	Relationship Level (Actual)
MS Flowrate Increasing	0.9	0.0
RCS Pressure Decreasing	0.9	0.9
RCS Temperature Decreasing	0.9	0.0

These symptom sets resulted in the following initiating event diagnosis (Table 7 and Table 8):

Table 7 - Spray Valve Diagnosis Results (Perceived)

Event Category	Initiating Event	Relationship Value	
		Lower Bound	Upper Bound
Normal Operation	Controller Failure - RCS Pressure	0.9	1.0
	Controller Failure - RCS Temperature	0.0	0.4
	Changing Steam Demand (increase)	0.0	0.7
AOO	Leak – MS System	0.0	0.6

Table 8 - Spray Valve Diagnosis Results (Actual)

Event Category	Initiating Event	Relationship Value	
		Lower Bound	Upper Bound
Normal Operation	Controller Failure - RCS Pressure	0.9	1.0
	Controller Failure - RCS Temperature	0.0	0.1
	Changing Steam Demand (increase)	0.0	0.1
AOO	Leak – MS System	0.0	0.2

In both the perceived and actual symptom cases, an RCS pressure controller failure was identified as the most likely event. An increase in steam demand was also identified as a possible event diagnosis for the perceived set of symptoms. Increasing steam demand is categorized as a normal operation event and is typically associated with operator actions to maintain the balance between reactor power and steam demand. Therefore, an operator believing this diagnosis might be expected to increase reactor power (by withdrawal of control rods) to balance the increased steam loading. Furthermore, the belief that an accident has not occurred could bias an operator to perform actions that would expedite a return to normal operations (such as bypassing an engineered safety feature to prevent an unnecessary transient). In this case, the inference method not only provides a rational basis for the operator's unsafe actions during the event, but also correctly identifies a pressure control system failure (i.e., a stuck open spray valve) as the actual event cause.

5.5.3 Application of the Diagnostic Process

In the current version of ADS-IDAC, the diagnostic engine supports three functions: (1) the identification of an abnormal event requiring the initiation emergency operating procedures (EOPs), (2) the activation of knowledge-based actions based on the identification of mass, energy, or momentum flow imbalances in the reactor plant, and (3) the determination of a component's relevance to the operator's situational assessment. The need to initiate EOPs is determined by comparing the maximum membership value for anticipated operational occurrences and accidents to a pre-defined emergency threshold specified in the operator's knowledge base. When the emergency threshold value is exceeded, the operator will initiate the EOPs. By varying the magnitude of the operator's emergency threshold (see Section 7.2.2), the analyst can adjust the time lag between event initiation and start of the EOPs. When the diagnostic engine calculates a high membership value for a plant functional imbalance, a set of knowledge-based actions are activated to restore the functional balance. Knowledge-based actions are described in greater detail in Section 5.6. Finally, during the execution of procedural actions, the ADS-IDAC model compares the functions performed by the component referenced in the procedural step to the operator's functional imbalance diagnosis. If the component is associated with an identified functional imbalance (e.g., the opening the pressurizer spray valve when the operator has diagnosed a high energy condition in the pressurizer), the action is assigned a high relevance score. The relevance score is used to determine the probability of skipping a procedural step and is discussed in more detail in Section 6.

5.6 Knowledge-Based Actions

A key goal of the current ADS-IDAC research effort is modeling knowledge-based actions that an operator might perform outside the scope of the emergency procedures. The plant functional decomposition map and diagnostic engine described in Section 4.3.1 provide a means to identify situations where operators might execute actions they believe to be reasonable given their situational assessment but are not necessarily covered by plant procedures. Examples of such actions include reducing reactor coolant system water injection when pressurizer level is high or decreasing the steam dump rate when steam generator pressure is low [85]. Within ADS-IDAC, knowledge-based actions can be activated when, based on the operator's perceived plant state, the event membership value of a functional imbalance diagnosis exceeds a pre-defined threshold value. For example, an imbalance diagnosis of "low mass in the reactor coolant system" might lead an operator to increase reactor coolant system injection flow, reduce normal letdown flow, or actuate emergency core cooling systems. Knowledge-based actions have the following characteristics and properties in the ADS-IDAC model:

- Action rules are organized within functional imbalance diagnostic groups. Each possible functional imbalance event can be associated with a list of actions intended to mitigate the associated mass, energy, or momentum imbalance.
- Each functional imbalance diagnosis group is assigned a priority level in order to reflect the relative importance of the associated actions to the operator. For example, actions intended to address inadequate core cooling might be sequenced before actions to address low steam generator inventory in a single steam generator. The priority can be adjusted to reflect an operator's knowledge, experience, and problem solving style.
- Each action can be assigned a set of prerequisite conditions that must be met prior to execution of the action. Prerequisites are used to better model the heuristic rules an operator might use to activate a specific action. Prerequisites can be associated with plant parameters, component states, alarms, active procedures in use, and an operator's mental beliefs.
- Once an action in a functional diagnosis group has been activated, further actions within the functional area will be blocked for a pre-defined dormancy period. The dormancy period allows the operator to address other, possibly lower priority, functional areas.

Because the activation of knowledge-based actions depends on the information perceived by the operator, information filtering, distortion, or biases may impact when action rules are activated. This can lead to human error events such as the execution of a knowledge-based rule during an inappropriate situation [30].

Application Example

As an example of the implementation of knowledge-based actions, consider the mitigative actions associated with a steam generator tube rupture. The EOPs direct the operator to isolate the affected steam generator and reduce primary pressure in order to equalize reactor coolant system pressure and ruptured steam generator pressure. This action reduces coolant leakage through the ruptured steam generator tube and facilitates refill of the reactor coolant system. Although a knowledge-based paradigm for execution of this action may not match the efficiency and stability afforded by the EOPs, knowledge based rules can be used to achieve a similar end state. Table 9 provides an overview of the functional diagnoses and associated knowledge-based actions that could achieve depressurization following a steam generator tube rupture. The detailed knowledge-based model used for this scenario is provided in Appendix F.

Immediately following the initiation of the steam generator tube rupture, the operators would likely note a decreasing mass condition in the reactor coolant system. Based on low pressurizer level, a knowledge-based action to actuate emergency core cooling would be activated. Should the operator identify either a high steam generator radiation condition or an uncontrolled level increase in the ruptured steam generator, a mental belief that a steam generator tube has occurred would be formed. This mental belief would satisfy other action prerequisites in Table 9 and lead the operator to depressurize the reactor coolant system and isolate the ruptured steam generator. If these actions were successful, the operator would eventually identify a high mass condition in the pressurizer and terminate emergency core cooling

Table 9 - Knowledge-Based Action for a Steam Generator Tube Rupture

Imbalance Diagnosis	Actions	Prerequisites
Low mass in the reactor coolant system	Isolate chemical and volume control system letdown	• Low pressurizer water level
	Actuate emergency core cooling systems	• Low pressurizer water level
High mass in the reactor coolant system	Stop emergency core cooling injection flow	• Adequate sub-cooling margin and pressurizer water level
Low energy in the pressurizer system	Reduce reactor coolant system temperature by decreasing the setpoints of the SG atmospheric relief valves	• Operator goal to reduce reactor coolant system temperature and low subcooling margin
High mass in a steam generator	Isolate steam generator (e.g., shut main steam, main feedwater, and auxiliary feedwater isolation valves	• Mental belief that reactor is shut down and a steam generator tube rupture has occurred.
	Open pressurizer spray valve	• Mental belief that reactor is shut down and a steam generator tube rupture has occurred. • RCS pressure ***greater*** than ruptured SG pressure
	Close pressurizer spray valve	• Mental belief that reactor is shut down and a steam generator tube rupture has occurred. • RCS pressure ***less*** than ruptured SG pressure

providing that adequate sub cooling margin existed. Other supporting actions, such as cooling the reactor coolant system to support depressurization, could also be identified in the operator's knowledge base.

To illustrate this example, a simulation was conducted for an accident involving a main steam line break (MSLB) followed by an induced steam generator tube rupture. Two different mitigative strategies were investigated: a strict procedure following approach using the Halden emergency operating procedures for this event, and (2) a knowledge-based approach using the rules outlined in Table 9. The response of pressurizer pressure and minimum subcooling are presented in Figure 17 and Figure 18. Both knowledge-based and procedurally driven approaches maintained a sufficient safety margin to prevent a core damage event from occurring. In particular, both approaches essentially reached the same end state, with reactor coolant system pressure approximately equal to steam generator pressure (thus terminating the SGTR event) and a adequate degree of subcooling. The main distinction between the two approaches is that the procedurally driven strategy initiated a reactor coolant system cool down prior to depressurization. This established a significant minimum subcooling margin prior to the depressurization. Conversely, the knowledge driven approach focused on termination of the loss of reactor coolant through the steam generator tube rupture and therefore a depressurization of the reactor coolant system was initiated earlier in the event. For the knowledge driven approach, reactor coolant system cooldown was driven by the impending loss of subcooling margin; consequently this approach had a much smaller margin to loss of subcooling than the procedurally driven method. Although this example is not intended to make any representation about how an actual crew would utilize knowledge-based actions, it does demonstrate the feasibility of building a knowledge driven model in the ADS-IDAC environment. Furthermore, it highlights how the strategy selection process can lead to very different operator interactions with the reactor plant.

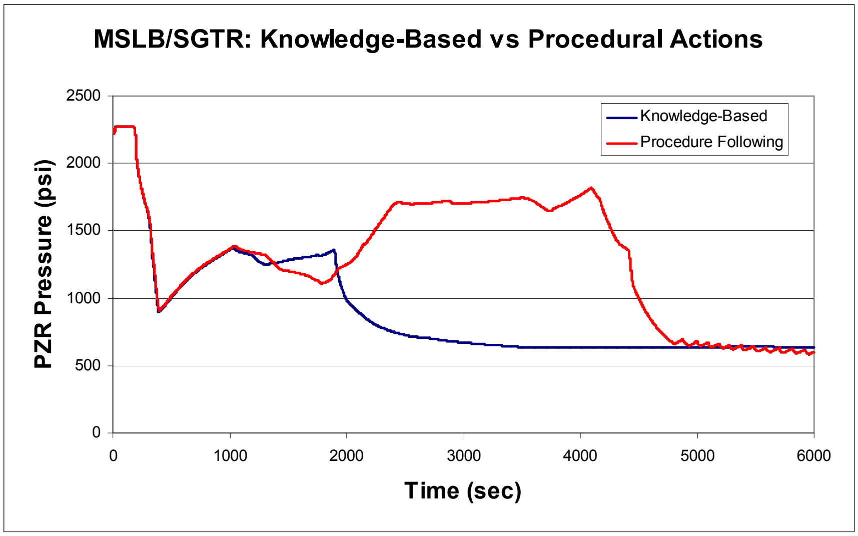


Figure 17 - Pressurizer Pressure Response

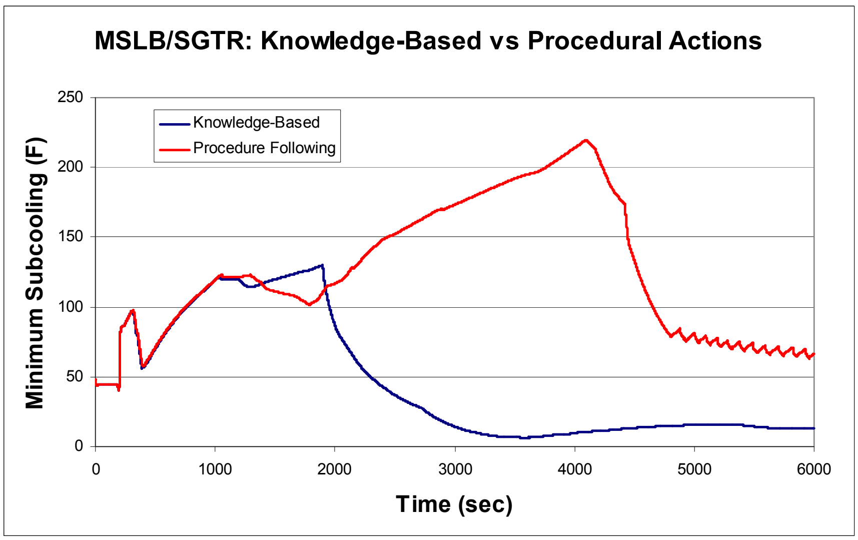


Figure 18 - Minimum Subcooling Response

6. Action Execution Module

The third and final stage of the IDAC cognitive model is action execution. Within ADS-IDAC, action execution refers to active interactions with the reactor plant model that alter the operating state of a plant components, directed information gathering, and the activation of procedure steps. The type of action and manner in which it is executed are dependent on the operator's action-response mode. Based on the results of the decision-making process, the operator model will select one of the following action-response modes:

- Actions specified in formal written procedures (e.g., emergency operating procedures);
- Skill- or rule-based actions arising from the activation of a mental belief; and
- Knowledge based actions based on the identification of a specific plant functional need.

In general, actions associated with written procedures and skill- and rule-based responses can include active component state changes, information gathering, and activation of new procedure steps. Actions resulting from the knowledge-based action-response mode are restricted to active component state changes and some limited information gathering activities.

Similar to an actual control room, ADS-IDAC provides four possible control inputs for each active component that can be manipulated by the control room crew:

1) changing the component operating mode (e.g., automatic vs. manual mode);
2) setting a specific control value for a component (e.g., throttling a control valve to 50% open);

3) incrementally adjusting the control setting of a component (e.g., throttling open a control valve by an additional 10%); and
4) setting or adjusting the setpoint of an automatic controller (e.g., setting the steam dump target pressure equal to the perceived main steam header pressure).

These capabilities provide sufficient flexibility to realistically model all significant active operator interactions with the plant model.

Many procedure steps are not associated with active changes to component operating states, but are instead intended to gather information about the operating condition of plant components or parameters. Therefore, ADS-IDAC action execution model also supports information gathering and verification. Information gathering can be conducted as single discrete event or can take place on a recurring basis. One time information gathering typically is used to verify the status of a parameter that is not routinely monitored, such as verifying that emergency core cooling systems started on demand. Recurrent monitoring is usually needed to monitor that state of a parameter that provides significant information about plant safety and may be used to activate additional follow-up activities. For example, some procedures may require the operators to monitor water levels in the pressurizer or steam generators and require specific follow-up actions if the level decreases below a pre-established threshold. Recurrent monitoring is performed by adding the parameter of interest to the operator scan queue (see Section 4.1.2). It should be noted that all information gathering activities are subject to perception processes and may result in the collection of biased or filtered information. In particular, operators with a tendency to rely on memorized information may not re-verify the status of a plant parameter if it has been previously perceived. Items contained in an operator's control panel scan queue may be removed if the scan queue becomes too large and the information is not perceived to be relevant based on the operator's situational assessment.

The final type of action execution is the activation of procedure steps. Skill- and rule-based memorized procedures are activated when the prerequisite conditions for the associated mental belief are met. Formal written procedures can be activated either by the operator crew detecting an accident condition or when triggered by a memorized procedure. Once the crew initiates formal written procedures, they may transition among several plant procedures to mitigate the accident event. In order to ensure proper executive control of the event simulation, certain rules are enforced for allowable procedure activations:

- Both the action-taker and the decision-maker can activate memorized procedures. Because only the action-taker can interact with the control panel, the decision-maker may require the action-taker's assistance to complete a memorized procedure.
- Only the decision-maker can follow a formal written procedure. This is consistent with nuclear plant standard practices that the senior reactor operator directs emergency operating procedures. This also has certain advantages in achieving adequate executive control over the simulation process.

A more detailed discussion of the implementation of plant procedures is provided in Section 4.3.3.

6.1 Performance Variability

Even when personnel selection, training programs, and administrative programs are consistently implemented, nuclear plant operators can exhibit significant crew-to-crew performance variabilities. These performance variabilities can arise from differences in crew knowledge, skills, and experience; crew specific organizational factors; and operator preferences and tendencies. If the sources of crew-to-crew variability can be modeled within a human reliability analysis, the prediction and mitigation of nuclear plant operator errors can be improved. ADS-IDAC provides a several types of branching events that can be used to model typical crew-to-crew variabilities:

- Action control value branches – For procedure step actions associated with a quantitative control input (e.g., opening a throttle valve to 10% open), two or more branches can be generated to explore the effect of variations in the control input. This branching rule can be used to model variations in how control room operators control certain transient events during operations (e.g., reactor coolant system cooldown and depressurization rates).
- Action time branches – Every operator action is associated with a probability distribution that describes the time required to accomplish the action. In the current version of ADS-IDAC, a three parameter Weibull probability distribution

is used to model the variability in action execution time. Activation of this branching rule allows the analyst to generate two or more sequence branches, each of which represents a sample taken over a different partition of the action time probability distribution.

- Information Biasing – Information read from the control panel can be biased by the operator's perception process, resulting in the collection of inaccurate plant data. For each operator, ADS-IDAC allows the analyst to specify instrumentation that is subject to biasing along with the type of biasing factor that is applied (e.g., additive, proportional, etc.). Information biasing is discussed in more detail in Section 4.2.
- Step-Skipping – Every procedure step includes several parameters to specify the likelihood that the step may be omitted during the action execution stage. The procedure step-skipping module is described in Section 6.2.

Prior to each ADS-IDAC event simulation, the analyst must specify what branching rules can be activated. This allows a thorough exploration of potential crew variabilities without an excessive generation of branching events and a sequence exploration condition. Section 9.2.1 provides a general framework for identifying appropriate branching rules for an event scenario.

6.2 Procedure Step-Skipping

Procedures in ADS-IDAC follow a standard three part format consisting of an action, one or more conditions that should result from the action, and contingency actions that should be performed if the expected conditions are not met. This is consistent with a standard "action," "action expectation," and "response not obtained" format commonly used at some nuclear plants. An important aspect of this modeling approach is that ADS-IDAC is capable of capturing not only the content, but also the structural format of the plant procedures. Because procedures are structured such that the contingency actions are only performed if the action expectations are not met, the expectations occasionally use unusual phrasing. For example, an expectation intended to verify that the steam generators are intact might be worded "no steam generator pressure decreasing in an uncontrolled manner." The interpretation of this condition would likely be influenced by the operator's biases and their situational assessment. ADS-IDAC is capable of representing these type of complex action expectations.

ADS-IDAC supports the modeling of omission of certain procedure actions in order to model step-skipping behavior. In order to provide adequate executive control over the simulation, step-skipping behavior is limited to the initial step actions and the contingency "response not obtained" actions. Although, ADS-IDAC cannot currently model the skipping of whole procedure sections (e.g., jumping from step 5 to step 15), if the steps within a section are subject to the same dependent factors, the model can generate sequences where all section steps are skipped. The likelihood of skipping a sub-step is calculated by adjusting a base probability value by dynamic and static multipliers. These multipliers reflect procedural characteristics, the relevance of the action to the operator's situational assessment, and the state of performance influencing factors.

6.2.1 Static Factors

Static factors refer to the properties of the procedure and are not expected to change during the accident event. In ADS-IDAC, static factors include procedure type, step objectives, and step complexity. These factors are summarized in Table 10.

Table 10 - Procedure Step-Skipping (Static Factors)

Static Factor	Example
Type of Procedure	• Normal • Abnormal • Optimal Recovery Guidelines (EOPs) • Functional Recovery Guidelines (FRGs) • Alarm Response • Memorized Actions (Skill of the Craft)
Type of Step	• Verification Steps • Prerequisite Steps • Decision Making Steps • Objective-Related Action Steps • Monitoring Steps
Complexity of Actions	• Location of Actions • Step Structure • Familiarity with Action

In the U.S., quality assurance program requirements require that plant operators to specify the manner in which procedures are to be executed [86]. The methods used by operators to execute procedures can vary depending on the type of procedure. Routine procedural actions that are frequently repeated may not require the procedure to be present. Conversely, procedures covering infrequent or complex tasks should normally be present at the job site and followed. Six procedure types are considered: normal operating, alarm response, abnormal, emergency optimal recovery guidelines, emergency functional recovery guidelines, and mental (skill of the craft) procedures. Each procedure type is assigned a factor from 1 to 10 to reflect the procedural adherence tendencies of the operators (with high values indicating a lower adherence tendency).

The objectives of procedure steps may also affect the operator's adherence tendency. For example, steps that are clearly aligned with the high level objectives of a procedure are unlikely to be skipped while monitoring or verification activities might be more likely to be missed. ADS-IDAC uses the following five categories to group step objectives: monitoring, prerequisite, verification, objective-related, and diagnosis-related steps. Monitoring steps require the operator to periodically check the value of a parameter or condition while verification steps require the operator to ensure that an expected condition exists. Prerequisite steps support later actions but are not directly associated with the high level goals of the procedure. Objective-related steps are directly associated with the high level goals of the procedure. Diagnosis steps require the operators to assess the plant state and possibly transfer to a new procedure path. Similar to the procedure type, the analyst assigns a factor from 1 to 10 to reflect the operator's tendency to skip these various step types.

The complexity of the procedure step is also considered a static factor. Complexity can refer to the step structure, the type of action, and the presence of actions inside and outside of the control room. Similar to the static procedural factors, the static step complexity factor ranges from 1 to 10, with a higher value reflecting a greater tendency for action skipping. Based on recent research that indicates steps with intermediate complexity may have the lower adherence that simpler or more complex steps [87], this factor reflects the likelihood of adherence due to step complexity rather the actual step complexity. The three static factors (procedure type, step objective, and complexity) are multiplied together to provide an overall static factor (f_{static}) for step-skipping.

6.2.2 Dynamic Factors

Two types of dynamic factors are used to adjust the basic step-skipping probability: (1) performance influencing factors, and (2) the relevance of the action to the operator's situational assessment.

Table 11 - Procedure Step-Skipping (Dynamic Factors)

Dynamic Factor	Description
Time Constraint Loading	• Based on time until a critical parameter exceeds a specific threshold • Represents plant dynamics
Action Relevance	• Measures the relevance of the step action to the operator's situational assessment • Action is closely associated with an identified plant need have high relevance

Because high time pressure may influence an operator's tendency to skip procedure steps, the time constraint loading performance influencing factor (PIF) is included in the step-skipping model. The time constraint load PIF varies in the range of 1 to 10, with a higher value indicating increased time pressure. The relevance of an action to the operator's situational assessment is determined by comparing the functions of the component references by the action to the output from the diagnostic engine. The plant component functional map (Section 4.3.1.3) specifies all functions supported by a component. Further, each function is directly associated to an imbalance event included in the diagnostic engine. Based on information perceived by the operator, the diagnostic engine calculates a membership value for each imbalance diagnosis. A relevance score for each component action is then calculated by Equation 4.

$$R = 10^{1-2d} \qquad (Equation\ 4)$$

where R is the relevance factor and d is the maximum membership value of all functional imbalances associated with the action. Because d varies from 0.0 to 1.0, the relevance factor, R, varies from 0.1 for highly relevant actions to 10.0 for irrelevant actions (a low value of R is associated with a lower step-skip probability). Because the amount and accuracy of plant data perceived by the operator changes over time, the relevance factor is a dynamic quantity. An operator with an accurate situational assessment will be less likely to skip pertinent actions, while an operator with a poor situational assessment may skip important steps. Actions that are not associated with a specific component (such as procedure transfers) are assigned a relevance factor of 1.0. The action relevance factor (R) is multiplied by the time constraint load PIF[7] to yield the overall dynamic factor ($f_{dynamic}$).

6.2.3 Implementation

Based on the static and dynamic step factors, an adjusted step-skipping probability is calculated using Equation 5 [88]:

[7] Because the time constraint load PIF can range from 0 to 10, an adjusted PIF value is used to calculate the $f_{dynamic}$ factor. When the time constraint load PIF is less than 1, the relevance factor is multiplied by $10^{(PIF - 1.0)}$, otherwise, the PIF factor is used directly. This results in a $f_{dynamic}$ range of 0.01 to 100.

$$P_{Skip} = \frac{P_{Base}\ f_{Static}\ f_{Dynamic}}{[P_{Base}\,(f_{Static}\ f_{Dynamic} - 1) + 1]} \qquad (Equation\ 5)$$

where P_{base} is the basic step-skipping probability and P_{skip} is the adjusted probability. The dynamic calculation of the step-skipping probability provides a number of advantages, including: (1) the ability to consider procedure type, step intent, and step complexity, (2) the influence of time pressure, and (3) the ability to link step-skipping tendencies to the operator situational assessment through the relevance factor. An example application of the procedure step-skipping model is presented in Sections 10.1.1.2 and 10.2.2.

7. Performance Influencing Factors and Crew Model

Within the IDAC model, the crew model specifies the roles and responsibilities for each member of the control room crew, communication protocols, preferences and tendencies, and decision-making dependencies. Performance Influencing Factors (PIFs) are those elements that could alter to course of a plant event due to their effect on human response. Taken together, the ADS-IDAC crew model and associated PIFs provide a means to realistically simulate nuclear plant control room crew performance and to explore factors that could lead to error events.

7.1 Human Performance Influencing Factors

Operator and crew behavior is influenced by static and dynamic performance influencing factors (PIFs) that capture internal and external factors that can affect cognitive performance [20]. The IDAC model groups PIFs into eleven broad categories:

- Cognitive modes and tendencies – alertness and attention
- Emotional arousal - stress
- Strains and feelings – task and time loading
- Perception and appraisal – situation perception and awareness of roles
- Intrinsic characteristics – confidence and motivation
- Memorized information – knowledge, experience, and skills
- Organizational factors – work practices and tools
- Team-related factors – cohesiveness, coordination, and leadership
- Conditioning events – latent hardware, software, and human failures
- Environmental factors – physical access, lighting, temperature, etc.
- Physical factors – fatigue, physical limitations

Proper implementation of a PIF model provides a means for capturing the dependencies that affect individual operator and crew behaviors, such as procedure step-skipping and decision-making preferences.

ADS-IDAC employs both static and dynamic performance influencing factors (PIFs) to influence and shape operator behavior. Static PIFs are constant parameters intended to represent the fixed environmental and organizational factors that affect crew behavior. Conversely, dynamic PIFs reflect transient conditions and model variations in the operator's mental state during a scenario. Dynamic PIFs provide an important mechanism for providing transient feedback to the operator model. Although certain static PIFs might be expected to change over time (e.g., the impact of increased training effectiveness on crew performance), the main distinction between static and dynamic PIFs is the time scale over which these factors change. Because an ADS-IDAC analysis is generally limited to the initial phase of an accident scenario, any factor that does not change significantly over a few hours is considered to be static.

7.2 Static Performance Influencing Factors

Static PIFs include all the human behavior influencing factors that are considered to remain constant over the initial control room response to an accident or other abnormal event. The current implementation of ADS-IDAC is intended to simulate control room crew performance up to the point that emergency response facilities are fully staffed. Once emergency facilities such as the Technical Support

Center and the Emergency Operations Facility are staffed and operational (typically within a few hours after an accident initiator), the dynamics of crew decision-making and operator response changes as these facilities begin to manage and coordinate the accident response. In general, the ADS-IDAC static PIFs involve organizational factors, team-related factors, and intrinsic operator characteristics. Additionally, a number of dynamic PIFs rely on static parameters that influence how the dynamic factors are calculated. For example, information load is modeled with a dynamic PIF, but the formula used to determine the PIF value depends on several static factors that characterize the operator's information processing capabilities. The specific static parameters used in the calculation of dynamic PIFs are discussed in Section 7.3. Three broad categories of static PIFs have been implemented in ADS-IDAC: memory limitations and use, activation thresholds, and goal and problem solving strategy selection.

7.2.1 Memory Limitations and Use

The operator model includes a memory capability that allows the operator to store and use previously perceived information. This provides an enhanced ability to model contextual factors associated with the operator's recent experience. The memory model includes two basic static PIFs that characterize the tendency of the operators to rely on memorized information and the operators' information gathering and perception capacities. Both of these factors are associated with the memorized information PIF group in the IDAC model.

Use of Memorized Information

This static PIF parameter establishes the branching probability for enabling the operator's use of previously perceived (and memorized) plant data. When the use of memorized information is enabled, the operator will use previously perceived information stored in intermediate memory (if available and current) to address data requirements of procedure expectations and knowledge-based action prerequisites. When the use of memorized information is blocked, the operator will always obtain current information from the plant control panel. The use of memorized information can reduce activity execution time but may result in the use of outdated and incorrect information. This factor is set in the range of 0.0 to 1.0 and represents the branching probability that the operator will initially attempt to use information already present in memory. Intermediate values will cause a branching point to be generated early in the simulation where one branch enables the use of memorized information (with the branching probability set to the input value) and a second branch blocks the use of memorized information (with the branching probability set to the complement of the input value).

Even when the use of memorized information is enabled, the operator may block the use of previously perceived information if it has not been recently perceived. The likelihood that the operator will use previously memorized information is specified by probability density functions that define a memory residency time for information related to an alarm status, component state, and parameter value. Three separate three-parameter Weibull probability distributions (a total of nine separate parameters) are used to model each of these information categories. When the use of memorized information is enabled, the operator will check to determine if desired information has been previously perceived. If the information had been previously perceived, a Monte Carlo simulation is used to calculate a maximum information residency time. If the age of the perceived information is less than the maximum information residency time obtained from the Monte Carlo simulation, the operator will use the memorized information. If the age of the perceived information is greater than the maximum residency time, the operator will obtain the current alarm state from the control panel.

Short Term Memory Capacity – Control Panel Scanning

The operator control panel scanning model (see Section 4.1.2) dynamically focuses operator attention on the parameters, component states, and alarms that the operator believes are most pertinent to the perceived plant state. The maximum number of items that can be monitored by the scan queue is established by dynamic limits based on the status of the information load and system criticality dynamic PIFs. This maximum scan queue limit is specified individually for each operator and is considered to be a static PIF. A higher scan queue limit will allow the operator to monitor more parameters and obtain an improved situational assessment of the plant state. Setting a lower scan queue limit reduces the number of parameters that can be periodically monitored. This flexibility allows the analyst to simulate the operator's information processing and short term memory limitations. When the size of the scan queue exceeds the dynamic limit, lower priority parameters are removed from the scan queue until the size limitation is met. Thus, an operator with a lower scan queue limit will have greater difficulty monitoring sufficient information to maintain an accurate situational assessment.

7.2.2 Activation Thresholds

Several key features of the operator decision-making model are linked to situations in which a critical parameter is exceeding a pre-determined cutoff value or activation threshold. As discussed in Section 2.5.1.1, the Cutoff Reinforcement Learning Model assumes that a decision maker will establish and adjust an activation cutoff value based on a learning process over a series of repeated trials. The learning process model includes factors such as positive reinforcement, generalization, and forgetting. Inspired by this model, ADS-IDAC includes static PIFs that establish critical activation thresholds. Although the Cutoff Reinforcement Learning thresholds are dynamic, over the time span of interest in an ADS-IDAC simulation, it is reasonable to assume a static threshold. However, if desired, these PIFs can be adjusted to reflect changes in an operator's experience, confidence, or tendencies to ignore or act upon information associated with a degrading plant state. Two primary categories of thresholds are currently implemented in ADS-IDAC – the detection threshold for an abnormal plant condition and the probability threshold to activate procedure step-skipping behavior

Detection of Abnormal Condition

This parameter establishes the diagnostic threshold for detection of an abnormal condition. As described in Section 5.5, the ADS-IDAC model includes a fuzzy logic diagnostic engine that supports the operator's situational assessment of the plant state. Three major classes of plant events are included in the diagnostic process: (1) normal operating events, (2) anticipated operational occurrences, and (3) design basis accidents. Based on the information perceived by the operator, the diagnostic engine calculates a membership value for each possible diagnostic event. These membership values range from 0.0 to 1.0 and represent the degree of matching between the event symptoms and the symptoms that have been perceived by the operator. Each time step, ADS-IDAC recalculates the maximum membership value of all events within the anticipated operational occurrence and design basis event categories. If this maximum membership value exceeds the abnormal signal threshold, the operator will conclude that an abnormal condition exists. The abnormal condition diagnosis does not "reset" once it is activated, even if the maximum membership values for anticipated operational and design basis events later decrease below the threshold. This parameter effectively establishes the operator's sensitivity to detecting an abnormal event. If the parameter is set to a high value, the operator will require more information to support an abnormal condition diagnosis. A lower value reduces the information requirements for detecting an abnormal condition, but might result in the operator reaching a "false positive" conclusion for an abnormal event. In practical terms, setting a high value tends to delay the operator's perception of an abnormal event following an accident initiator, while a lower value tends to support earlier activation of this perception. The determination that an abnormal condition exists influences the goal selection process and may lead to activation of the emergency operating procedures. This threshold value can be used to characterize the operator's self confidence level (an intrinsic characteristic in the IDAC model) or the operator's perceived severity of the consequences associated with their diagnosis (a perception and appraisal characteristic).

Following the identification of an abnormal condition or the perception of a reactor trip, the operator will suspend all low priority mental procedures. The priority of a mental procedure is represented by an integer value of 1 or greater. High priority procedures are associated with a lower priority value (i.e., the highest priority procedures have a priority value of "1"). The profile for each operator includes a static PIF to define the priority threshold for mental procedures. Once an abnormal condition has been detected, the operator suspends the execution of all active and queued mental procedures with a priority level lower than the specified threshold value. For example, if the priority threshold is set at 2, all mental procedures with a priority value of 3 or greater will be suspended. The purpose of this parameter is to allow the operator to interrupt mental procedures that are no longer appropriate following a reactor trip or while mitigating an accident event. Because the abnormal condition diagnosis is not reset, the suspension of low priority mental procedures can occur only once during an accident sequence. Thus, low priority mental procedures that are activated following the initial diagnosis of a reactor trip or abnormal event are not automatically suspended and may be executed.

Procedural Adherence

Procedural steps in ADS-IDAC have three main components: (1) initial action activity, (2) expectations associated with the initial action activity, and (3) a non-response action that is executed if the action expectations are not met. The operator may skip either the initial action activity or the non-response action. Evaluation of the action expectations cannot be skipped. ADS-IDAC dynamically calculates the probability of skipping each procedure step executed by the operator (see Section 6.2). If the calculated step-skipping probability exceeds a preset threshold value, two dynamic event tree branches are generated – one branch executes the associated procedure step while the second branch skips the step. The branching probability for the step-skipping branch is set equal to the calculated branch probability, and the branch probability for execution of the step is equal to the complement of the skip probability. If the calculated skip probability is less than the skip action threshold, the associated procedure step action is executed and no branching point is generated. Thus, the procedure step-skip action thresholds set the minimum probability for generating a branching point for skipping the initial and non-response actions for a procedural step.

The skip threshold was originally intended simply to suppress the generation of excessive numbers of low probability branching points. Although sequence truncation limits would serve a similar purpose, these limits are not applied until the branching point has been generated, and it is more computationally efficient to suppress low probability branches before they are generated. The analyst can reduce the excessive generation of procedure step-skipping branches by increasing the skip action threshold. Use of the skip threshold also provides the analyst an additional degree of freedom in specifying procedural adherence tendencies of the control room crew. In this manner, the thresholds can be considered a static PIF that characterizes the operator's tendency for attentiveness to the current task (a PIF grouped under cognitive modes and tendencies in the IDAC model). Setting the skip thresholds to higher values can be used to model crews with a higher likelihood to adhere to procedural requirements (e.g., setting the thresholds equal to 1.0 will suppress all skipping behavior).

7.2.3 Goal and Problem Solving Preferences

ADS-IDAC includes two static PIF factors to represent the operator's preferences regarding goal and strategy selection. The troubleshooting probability and procedure-use probability PIFs support the goal selection process and influence the crew's selection of problem solving strategy. These PIFs are related to the operator's intrinsic characteristic PIF group in the IDAC model and characterize the crew's problem solving style.

The troubleshooting probability establishes the branching probability that the decision-maker will activate the troubleshooting goal when plant parameters indicate an abnormal condition (see Figure 15). When the troubleshooting probability is set to 1, the decision-maker will always bypass the monitoring goal in favor of troubleshooting. A value of 0 will cause the decision-maker to bypass the troubleshooting goal in favor of monitoring the situation. For intermediate values, a branching point will be generated with two operator goal branches. One branch will activate the troubleshooting goal with a branch probability equal to the troubleshooting probability. The other branch will activate the monitoring goal with a complimentary branching probability. When the troubleshooting goal is activated, the crew can implement knowledge-based actions to address the abnormal condition (see Figure 16). This PIF parameter is currently implemented only for the decision-maker.

The procedure-use probability establishes the branching probability that the decision-maker will transition from the troubleshooting goal to the maintaining global safety goal (see Figure 15) after a reactor trip has occurred. When the procedure-use probability is set to 1, the decision-maker will always abandon the troubleshooting goal in favor of the maintaining global safety goal when a reactor trip condition is detected. A value of 0 will cause the decision-maker continue activation of the troubleshooting goal regardless of the reactor status. For intermediate values, a branching point will be generated with two operator goal branches. One branch will activate the transition from the troubleshooting goal to the maintaining global safety margin goal with a branch probability equal to the procedure-use probability. The other branch will block the transition to the maintaining global safety margin goal and therefore allow the crew to continue troubleshooting activities. Activation of the maintaining global safety goal enables the following-procedure problem solving strategy. Effectively, this parameter allows the crew to transition from a knowledge-based approach to accident mitigation to a procedure following approach. This parameter is currently implemented only for the Decision Maker.

7.3 Dynamic Influencing Factors

The ADS-IDAC simulation currently includes three dynamic PIFs: information load, time constraint load, and criticality of system condition. These PIFs serve as the main feedback mechanism between the thermal hydraulic model and the operator's mental state. The value for each of these dynamic PIFs ranges from 0 to 10. For these PIFs, a higher PIF value indicates a more adverse condition (e.g., higher information load, more time constrained situation, or degraded system condition). Information load is currently used as a surrogate measure of the operator's passive information load, task related load, and non-task related load – three items which are included within the strains and feelings PIF group in the IDAC model [20]. The criticality of system condition PIF factor is included under the perception and appraisal IDAC PIF group.

7.3.1 Time Constraint Load

The time constraint load PIF represents the time available until a monitored plant parameter exceeds a critical threshold. Because operators will normally monitor more than one important parameter, the overall PIF value is based on the most time limiting parameter. The knowledge base profile for each operator includes data defining how the time constraint load PIF value is calculated, including a listing of plant parameters used to calculate the time constraint PIF value along with the associated critical threshold values. Typical parameters that may be included in the calculation of the time constraint PIF include steam generator water levels, pressurizer water level, and reactor coolant system pressure. In order to model the potential dependence between operating mode and critical parameter threshold values, two different threshold levels are used to calculate the PIF value – a normal operation threshold and an accident threshold. When the operator’s high level goal is maintaining normal operation, monitoring, or troubleshooting an abnormal condition, the normal operation threshold is used. These goals are associated with at-power operation; therefore, time limitations would be expected to be driven primarily by the desire to maintain normal operation by avoiding a reactor trip.   If the operator switches to the goal of maintaining global safety margins (or if a reactor trip has occurred with the troubleshooting goal active), the time constraint PIF value is based on the accident threshold. The use of two different threshold values allows ADS-IDAC to capture an operator’s changing sensitivity to key parameters depending on the overall perceived plant condition. In general, the normal accident threshold is set to a level corresponding to reactor plant trip set points. The accident level threshold is normally set to a less restrictive value that is more indicative of the availability of a key safety function. For example, if a plant that has an automatic reactor trip on low steam generator (SG) water level, an operator might focus on the time available until the reactor trip set point is reached during an uncontrolled decrease in SG level. However, once the reactor is tripped, the operator’s focus might shift to simply maintaining adequate decay heat removal capability from the steam generator - a function that can be often be performed with a much lower SG level. Thus the normal threshold might be set equal to the low SG water level reactor trip set point while a less restrictive value is used for the accident threshold. Like all information processed by the operator model, the time constraint load PIF value is based on information perceived by the operator rather than data obtained directly from the thermal-hydraulic model. Perceived data may differ from the actual parameter value in thermal-hydraulic model due to time lags in updating perceived data and any distortions introduced by perception filtering and biasing.

The first step in determining the time constraint loading PIF is to determine the time available until each time constrained parameter exceeds a critical threshold (Equation 6).

$$t_{i,available} = \frac{P_i - P_{i,Threshold}}{\dot{P}_i} \qquad (Equation\ 6)$$

In Equation 6, $t_{i,available}$ is the time until the value of parameter i (Pi) exceeds threshold value $P_{i,Threshold}$ and $\dot{P}_i$ is the rate of change of parameter P_i. If an updated parameter value has not been perceived since the last PIF update, ADS-IDAC will extrapolate the parameter value based on the previously perceived parameter value and trend. To prevent this extrapolation process from artificially inducing an influence on operator behavior, when a parameter becomes the most time limiting factor, the scan queue (see Section 4.1.2) is verified to ensure the operator is actually monitoring the parameter. If the parameter is not being actively monitored by the scan queue, it is added to ensure that the limiting PIF value is based on perceived information rather than an extrapolation process. An updated PIF value associated with each monitored parameter ($PIF_{i,TimeConstraint}$) is then obtained using the value of $t_{i,available}$ from Equation 6 (see Equation 7a).

$$PIF_{i,TimeConstraint} = 10\left[1 - \frac{(t_{i,available} - t_{Lower})}{(t_{Upper} - t_{Lower})}\right] \qquad (Equation\ 7a)$$

The tuning constants t_{Lower} and t_{Upper} are used to calibrate the PIF value to the desired operator characteristics. Equation 7a is applicable only when $t_{i,available}$ is between t_{Lower} and t_{Upper}. If the minimum time available exceeds t_{Upper}, the time constraint PIF value is set to 0. If the minimum time available is less than t_{Lower}, the PIF value is assumed to saturate at a value of 10 (see Figure 19).

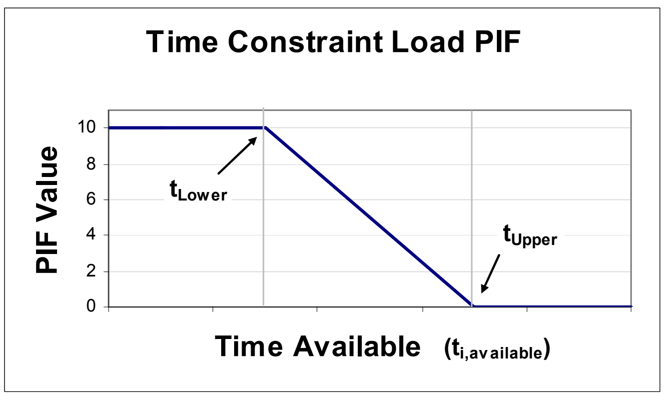


Figure 19 - Time Constraint PIF

In order to more realistically model dynamic changes in the time constraint PIF factor, the updated value of $PIF_{i,TimeConstraint}$ is passed through a lag filter to simulate the gradual buildup and decay of stress associated with time constrained loading. If the value of $PIF_{i,TimeConstraint}$ has not saturated at the maximum PIF value of 10, the lag filter output is determined from the following equation (Equation 7b):

$$PIF_{i,TCL}^{Updated} = PIF_{i,TCL}^{Old} + (PIF_{i,TCL}^{New} - PIF_{i,TCL}^{Old})(\frac{t_{lapse}}{\tau_{buildup}}) \qquad (Equation\ 7b)$$

Where:

PIF^{Old} is last updated PIF value;
$PIF^{Updated}$ is the output from the lag filter;
PIF^{New} is the updated and unfiltered PIF value obtained from Equation 7a;
t_{lapse} is the time constraint loading PIF update periodicity (in seconds); and
$\tau_{buildup}$ is the buildup time constant (in seconds).

Once the PIF value for a monitored parameter reaches saturation at the maximum value, the parameter PIF value is allowed to decay using the following formula (Equation 7c):

$$PIF_{i,TCL}^{Updated} = 10(1 - e^{(t_{current} - t_{threshold})/\tau_{decay}}) \qquad (Equation\ 7c)$$

Where:

$PIF^{Updated}$ is the output from the lag filter;
$t_{current}$ is the current simulation time (seconds);
$t_{threshold}$ is the time that the parameter exceeded the accident threshold value; and
τ_{decay} is the decay time constant.

This decay feature is intended to account for a decrease in induced time constraint loading once a parameter has passed a critical threshold, and the operator is unable to recover. In this case, the operator would likely focus on other critical parameters where mitigation actions may be more successful. The timing parameters can be adjusted by the analyst to match desired crew characteristics. The overall value for the time constraint PIF value is the maximum value of $PIF_{i,TimeConstaint}$ for all monitored parameters.

Example Application

In order to demonstrate the behavior of the dynamic PIF factors, an example steam generator tube rupture (SGTR) scenario was analyzed. The appropriate response to a SGTR event requires several dynamic interactions with the reactor plant, including initiation of emergency core cooling, isolation of the ruptured steam generator, cool down and depressurization of the reactor coolant system, and termination of emergency core cooling. Each of these steps either initiates or terminates a significant trend in a key reactor plant parameter. The thermal-hydraulic response for the event is shown in Figure 20. As can be seen in Figure 20, a slow pressurizer level decreasing trend begins immediately after the SGTR is initiated at approximately 180 seconds. The water level in the ruptured SG (SG A) remains relatively unperturbed during the initial phase of the accident because the SG water level control system compensates for the inflow of reactor coolant from the ruptured SG tube.

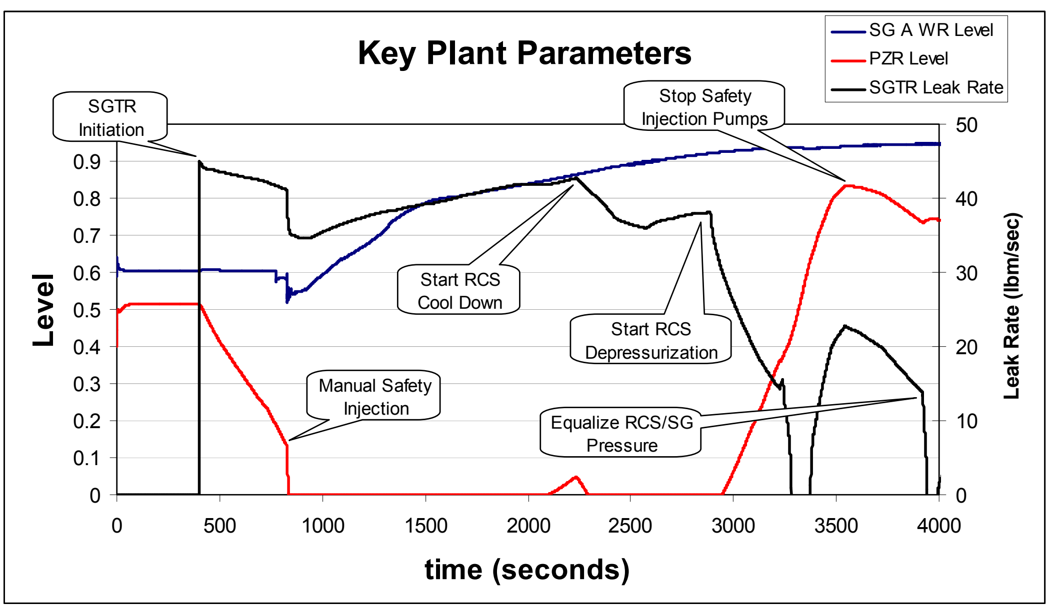


Figure 20 - Plant Response to SGTR

After the operators manually initiate the emergency core cooling system, the reactor plant shuts down, and the SG water level control system can no longer compensate for the reactor coolant leakage into the SG; consequently, SG A water level begins increasing at roughly 700 seconds. The net cooldown of the reactor coolant following shutdown causes the pressurizer water to decrease below the indicating range despite the high makeup flow from the emergency core cooling systems. As the operators work through the emergency operating procedure, they will initiate a cooldown and depressurization to equalize pressure across the SG tube and terminate the coolant leak. After the leakage into the ruptured SG is terminated, the pressurizer level increases rapidly, requiring the operators to terminate emergency core cooling injection before the pressurizer is completely filled. Once the emergency core cooling injection is stopped, the operators can re-equalize reactor coolant and SG pressure to stabilize plant conditions. Thus, the response to this accident causes significant transients on SG water level, pressurizer level, and reactor coolant pressure – all factors that would be expected to cause time constraint loading on the operating crew.

Figure 21 shows the response of the dynamic time constraint loading PIF to the SGTR accident. Shortly after the SGTR is initiated, the PIF value increases due to the pressurizer water level slowly decreasing toward the operational limit of 10%. After the reactor is shutdown, the time loading associated with pressurizer water level will slowly decay toward zero. However, the increasing SG water level will eventually trigger an increase in the time constraint loading as the steam generator

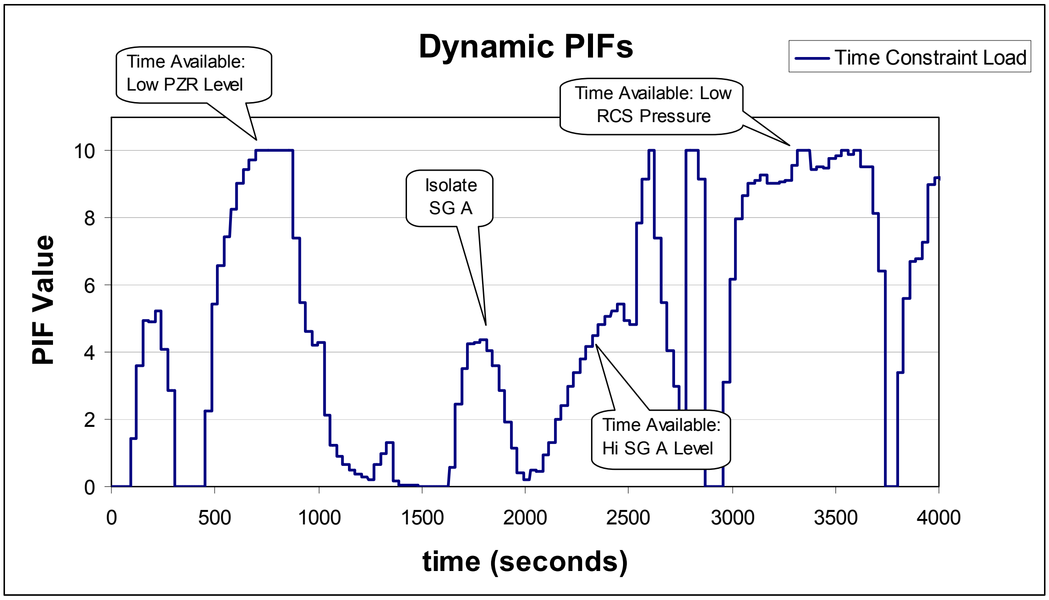


Figure 21 - Time Constraint Loading Response to SGTR

fills. Operator action to isolate other sources of makeup flow (e.g., auxiliary feedwater) into the steam generator slows the rate of level increase and temporarily reduces the time constraint loading associated with SG water level. The leakage of reactor coolant through the ruptured tube will continue to increase SG water level, and time loading associated with SG level will begin to build again a short time later. The time loading associated with SG water level will eventually saturate and begin to decay. Once the operators initiate reactor coolant system cooldown, the decrease in pressure will cause additional time loading – this is an expected condition since the operator must closely monitor the pressure decrease to ensure that adequate safety margins are maintained. Additionally, pressurizer level increases rapidly; this condition combined with the depressurization causes a period of relatively high time loading. The time constraint PIF varies in direct response to the operator's perception of adverse trends in key plant parameters and provides a representative measure of time available until a critical threshold is exceeded.

7.3.2 Information Load

The information loading dynamic PIF represents the operator's mental workload associated with the perception, processing, and communication of information. All information available from the nuclear plant thermal hydraulic model and crew communications must first pass through the operator's perception filter before it can be memorized and used. Consequently, the flow rate of information through the perception filter provides a convenient measure of each operator's information processing workload. The formula used to calculate the information load PIF is shown in Equation 8.

$$PIF_{Info\ Load} = 10\frac{(I_{Rate} - \alpha)}{(\beta - \alpha)} \qquad (Equation\ 8)$$

The variable I_{Rate} represents the operator's dynamic information processing rate and the constants α and β are calibration parameters. The calibration parameters can be adjusted to reflect an individual operator's information handling capabilities. Equation 8 is applicable only when I_{Rate} is between α and β. If the information processing rate is less than α, the PIF value is set to 0. If the information processing rate is greater than β, the information load PIF saturates to a value of 10 (see Figure 22).

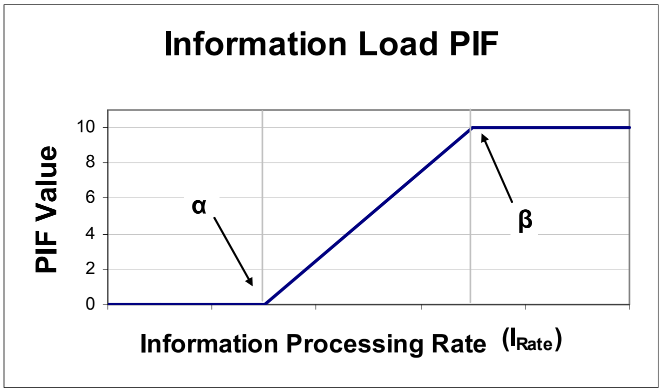


Figure 22 - Information Load PIF

Example Application

The dynamic response of the information loading PIF to the same accident scenario example described in Section 7.3.1 is shown in Figure 21. A notable

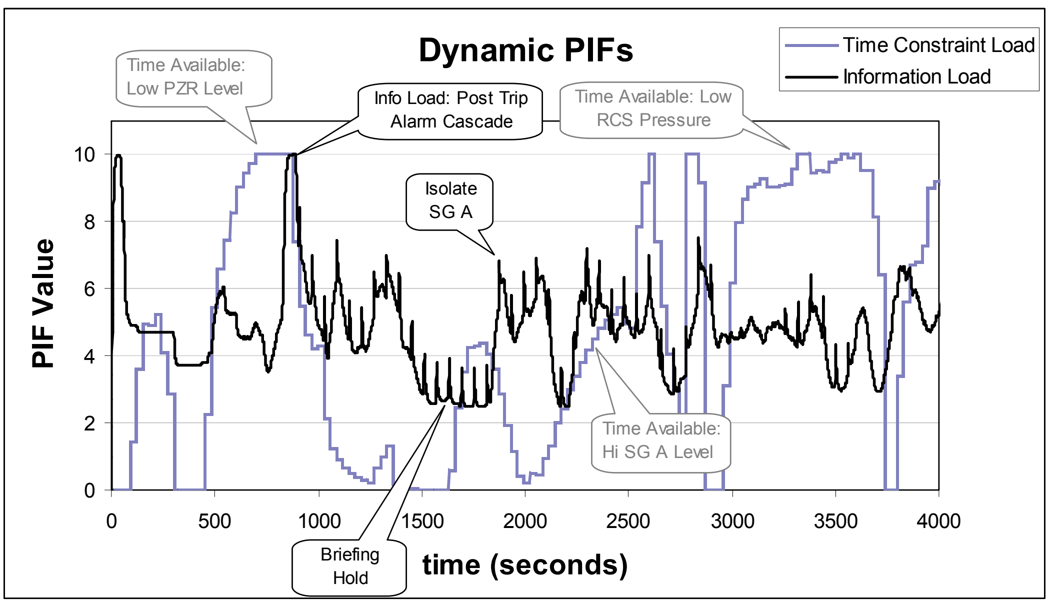


Figure 23 - Information Loading Response to SGTR

feature of the information loading PIF is that it appropriately tracks periods of high and low procedurally driven activity. Because the decision-maker directs all emergency operating procedures, actions requests and plant information must be continuously communicated between the crew members when procedure activity is high. For example, the PIF decreases during a briefing hold that occurs at approximately 1500 seconds – during this period, the operators are not executing any procedure actions, thereby reducing the operator's information processing load. Shortly after the briefing hold, the operators rapidly execute a series of actions needed to isolate the ruptured steam generator. This is denoted by a period of relatively high information loading around 2000 seconds. Also noteworthy is the information loading peak when the operators initiate emergency core cooling. The initiation of emergency systems and shutdown of the plant trigger numerous control room alarms that the operators must process, thus temporarily increasing the information processing rate. The information loading PIF provides another dynamic PIF that varies in response to real operator activity and plant conditions.

7.3.3 Criticality of System Condition

The criticality of system condition PIF represents the operator's perception of the level of degradation of key safety functions. This PIF is loosely based on the safety parameter display system used in U.S. nuclear plant control rooms [89]. The value of the system criticality PIF corresponds to the aggregate deviation of key safety parameters from a nominal value. Each operator profile identifies the parameters used to calculate this PIF, the threshold limits associated with each parameter, and the weighting factors used to aggregate the parameter contributions. Typical parameters used to calculate the system criticality PIF include reactor coolant system subcooling margin, wide range steam generator water levels, pressurizer water level, and reactor vessel water level. The contribution from each identified parameter to the overall criticality of system condition PIF value is denoted as the parameter criticality ($PIF_{Parameter\ Criticality}$). Given a set of high and low threshold limits, the parameter criticality corresponds to the magnitude of the parameter's deviation from a nominal "safe" condition (see Figure 24).

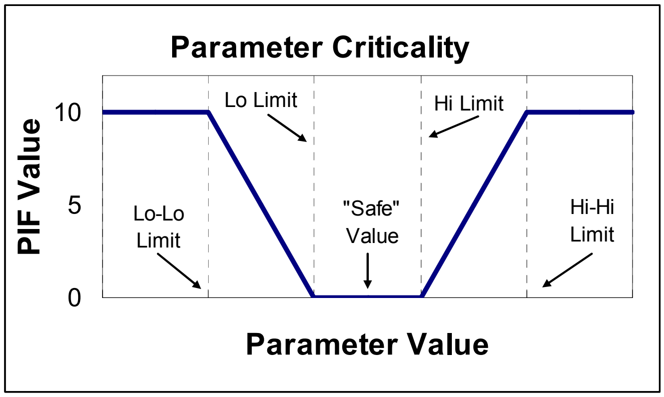


Figure 24 - Parameter Criticality PIF

As shown in Figure 24, the parameter criticality considers both high and low deviations from the nominal safe state. For example, a low level of reactor coolant system subcooling margin might indicate inadequate core cooling and an increased potential for core damage, while an excessive amount of subcooling might indicate an overcooling event and a potential pressurized thermal shock condition. The overall criticality of system condition PIF value is based on a weighted sum of the individual parameter criticality values (Equation 9).

$$PIF_{SystemCriticality} = \frac{\sum \omega_i \, PIF_{Parameter\ Criticality}}{\sum \omega_i} \qquad (Equation\ 9)$$

The ω_i value in Equation 9 is the weighting factor for parameter i. A higher value of the system criticality PIF indicates a more adverse overall plant condition.

Additionally, the rate of change of the PIF value provides an indication if the overall plant health is improving or worsening.

Example Application

The response of the system criticality PIF to the example SGTR accident scenario described in Section 7.3.1 is shown in Figure 25. This PIF factor

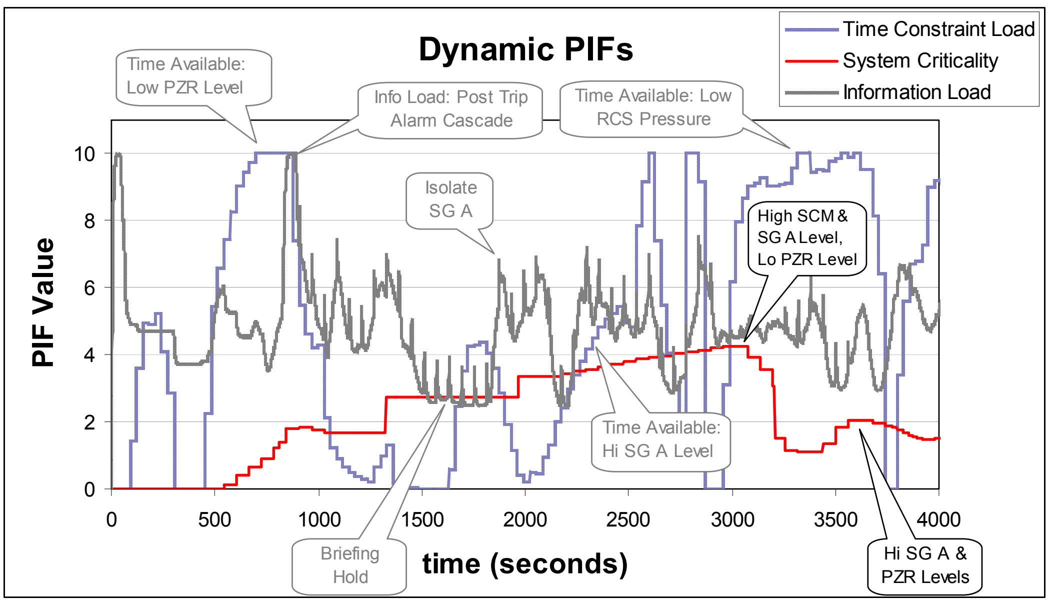


Figure 25 - System Criticality Response to SGTR

tends to have a more stable response to the SGTR accident that the time constraint and information loading PIF factors. Shortly after the initiation of the SGTR, the system criticality PIF begins to increase due to the deviation of pressurizer level from its normal operating range. As the operators initiate emergency procedures, the rate at which the system criticality PIF increases begins to level off, but the value continues to slowly increase until the operators take action to reduce reactor coolant system pressure and stabilize plant conditions. Because the water levels in the ruptured SG and the pressurizer are high and cannot be immediately corrected by emergency operating procedures, there is some residual value in the system criticality PIF near the end of the scenario. Similar to the time constraint and information loading PIFs, the system criticality PIF dynamically and realistically responds to changing plant conditions. Notably, the operator actions required by the emergency operating procedures are able to mitigate the increase in the system criticality PIF and return the PIF to a relatively low value.

7.4 Crew Communication and Coordination

The ADS-IDAC crew model currently includes a senior reactor operator (SRO) and a reactor operator (RO). Similar to an actual control room, each operator has unique roles and responsibilities. The SRO selects the high level goal and directs all written plant procedures. The RO performs all interactions with the nuclear power plant model through the ADS-IDAC control panel.

Certain crew activities require coordination and communication between the control room operators. For example, only the decision-maker can direct the performance of proceduralized actions and only the action-taker can manipulate the control panel. Therefore, the execution of a procedure step requires the decision-maker to direct the action-taker to perform the specified action followed by a report from the action-taker to the decision-maker that the action had been accomplished. The time required to perform inter-crew communication is established by the nominal communication time parameter specified in the profile for each operator. The parameter establishes the communication delay time (in seconds) and is controlled by the sender of the information.

The time required to perform an operator action is specified by the three-parameter Weibull probability distribution function in each mental or written procedure action step. However, it is sometimes desirable to globally change action time in order to model a fast or slow crew. Therefore, the operator profile includes a time multiplier factor to proportionally adjust the time required for the operator to execute activities. The action time multiplier is uniformly applied to all operator activities, including communication, action execution, and decision-making activities. A factor of 2.0 doubles the activity execution time compared to the baseline time while a factor of 0.5 reduces the activity execution time by a factor of ½. No dynamic event tree branches are generated by this parameter.

Within ADS-IDAC, the crew can interact with the plant model by manipulating components or gathering information from instruments and alarms. For computational convenience, all crew interactions with the nuclear plant model follow a standard sequence of events (termed the "action block" process). Because only the RO is permitted to interact with the ADS-IDAC control panel, any task initiated by the SRO must first be communicated to the RO. If the RO is not occupied with another task, the RO executes the requested action and communicates the status of the task back to the SRO. The SRO then compares the status report information to the expectations for the initial request and either: (1) proceeds to the next task request or

(2) performs a contingency action. If the RO is busy when the SRO initiates an action request, the request is held in an "ordered action" queue until the RO is able to execute the task. For instinctive response actions that are self-directed by the RO, the action block process is shortened to include only the RO's control panel interaction followed by a determination by the RO if the action expectation had been met. Consequently, self-directed actions by the RO can usually be accomplished more quickly than actions directed by the SRO.

To ensure that crew actions are coordinated, an order of precedence for problem solving strategies has been developed. The following rules guide the transition between operator problem solving strategies:

- The "Wait and Monitor" strategy has the lowest order of precedence and is only activated if no other problem solving strategy is active.
- The "Knowledge-Based Reasoning" and the "Follow Written Procedure" strategies are mutually exclusive and cannot be activated simultaneously.
- High priority "Instinctive Response" strategy actions will interrupt all other strategies. Lower priority instinctive response actions may interrupt other strategies depending on the crew's high level goal and the individual operator profile. Once the instinctive response actions are complete, the operator will return to the previous strategy.
- Transitions between problem solving strategies are not permitted when an action block is active. Once the current action block is completed, the operator may transition to a different strategy.

- The initial activation of the “Follow Written Procedure” strategy will automatically transition the operations crew to the specified emergency operations entry procedure (e.g., E-0). Subsequent activations of the “Follow Written Procedure” strategy will cause the operator to implement a new procedure (if specified) or return to the previous procedure and step.

These rules ensure the executive control of the simulation is well coordinated and crew activities are realistically modeled.

8. Discrete Dynamic Event Tree Generation

ADS-IDAC generates a dynamic event tree during accident scenarios by activating branching points when certain conditions are met. Each branching point includes two or more individual event branches, each of which represents distinct combinations of system and operator states. Collectively, the branching points describe the topology of a discrete dynamic event tree (DDET) associated with an initiating event. A specific accident sequence is defined by the unique path through the DDET branching points from the initiating event to an end state. The generation of branching points is controlled by a set of general rules that define the specific activation conditions for a branching point. Although the branching rules are predefined, the creation of branching points depends on the dynamic behavior of the reactor plant and operator decision-making models. Consequently, a simulation approach is needed to determine the branching points that are generated along a specific accident sequence trajectory. Because branching points can represent variability in crew actions, a calibration process has been developed to translate observed crew behaviors into relatively simple set of branching rules that can be applied within the rich contextual environment provided by ADS-IDAC. The calibration process is described in Section 9.2. This research study has demonstrated that relatively complex variations in crew-to-crew performance can be captured with a small set of generalized branching rules.

Because the branching rules guide the construction of the accident sequence DDET, these rules also define the scope of the ADS-IDAC analysis. However, a challenge of the DDET approach is that the size of the simulation state-space grows exponentially large as the number of branches increases. This phenomenon, known as sequence explosion, can limit the ability to complete an ADS-IDAC simulation within a reasonable time period. In general, branching rule definitions should capture a sufficient range of crew variability while minimizing the potential for sequence explosion. ADS-IDAC also allows the analyst to terminate an accident sequence when the sequence probability drops below a minimum threshold value, the sequence's simulated time limit exceeds a maximum limit, or a thermal-hydraulic parameter exceeds a specified threshold value. In practice, branching rules and sequence truncation criteria must balance the desire for full problem coverage with computing platform limitations.

Branching rules can be constructed to reflect variations in plant hardware and crew responses to plant events. By appropriately combining these branching rules, a wide spectrum of possible plant and operator states can be simulated. ADS-IDAC provides the capability to generate branching points to model variability in operator and plant hardware performance

8.1 Operator Performance Branching Events

A key research goal for the ADS-IDAC project is to develop a human reliability analysis tool capable of indentifying contextual factors that could lead to human error events. Therefore, a primary focus for event tree branching is capturing a wide range of potential human performance variability. ADS-IDAC currently supports branching events associated with all phases of the IDAC information processing, decision-making, and action execution processes.

8.1.1 Information Processing

The ADS-IDAC operator model is capable of using previously perceived plant information to support the decision-making process. For example, if a procedure expectation requires the operator to verify the state of a plant parameter, the operator can either actively gather information on the parameter from the control panel or rely on their memory of the parameter value if the information had been previously perceived. Use of memorized information can be more efficient since it reduces the time required to evaluate plant conditions. However, depending on the time delay between initially gathering the information and its eventual use, the operator's memorized data may not represent an accurate assessment of plant conditions. In order to allow exploration of this effect, each operator profile includes a PIF to enable the use of memorized information. The PIF can be used to toggle between information processing modes, or be set such that a branching point is generated early in the simulation to explore the impact of either using or not using memorized information. Only one branching point associated with the use of memorized information is generated during a sequence – once the branching rule is activated, all subsequent information processing will use memorized information as specified by the initial branching event. However, the branching rule is applied on an operator basis rather than a crew basis so crew members may use different information processing modes during a sequence.

8.1.2 Decision-Making

The ADS-IDAC decision-making process (described in Section 5) includes the ability to generate branching events for several key processes. These processes include goal and strategy selection and the activation of mental beliefs. Goal selection influences the high level problem solving approach used by the operating crew to address an accident event (e.g., use of written emergency operating procedures vs. knowledge-based troubleshooting). Mental beliefs represent high level observations, decisions, conclusions or ideas that the operator formulates regarding his/her assessment of plant status. Once a mental belief is activated, it can be used to activate other beliefs or drive the operator to perform skill- or rule-based actions or initiate a formal plant procedure.

8.1.2.1 Goal Selection

ADS-IDAC provides two branching rules to model variations in the crew's selection of a high level goal. The troubleshooting and procedure-use PIFs are included in each operator's profile and can be used enable certain goal selection preferences. These branching rules can be used generate two branches when they are activated: one that will enable either the troubleshooting or maintaining safety margins goals, and another that blocks activation of these goals. As described in Section 5.2, when the troubleshooting goal is activated, the crew may use non-proceduralized knowledge-driven actions to mitigate an accident event. Use of knowledge-based actions can result in more efficient accident mitigation since the crew is more likely to expend resources directly addressing adverse plant functions. However, this mitigation mode may also result in inappropriate actions if the operator's situational assessment is incorrect. If plant conditions continue to degrade during troubleshooting activities, the crew may transition to the more formalized procedure following problem-solving strategy depending on the setting of the procedure use PIF.

8.1.2.2 Mental Belief Activation

Every mental belief in an operator knowledge base includes an activation probability that characterizes the branching rule associated with the belief. When the prerequisite conditions for a mental belief are satisfied, two sequence branches can be generated – one branch where the mental belief is activated and another where the mental belief is left un-activated (or bypassed)[8]. Activation of a mental belief has two primary impacts: (1) enabling of skill- or rule-based follow-up actions, and (2) potential activation of prerequisite conditions for other mental beliefs. As described in Section 4.3.2, mental beliefs can be used to model complex non-proceduralized operator behaviors and form a significant portion of the operator knowledge base.

[8] To allow the analyst to prevent the excessive generation of sequence branches, activation probabilities set very close to 0.0 or 1.0 will only generate a single event sequence branch. Intermediate values will generate two braches with the split probability for mental belief activation set equal to the activation probability and the bypass probability set equal to the complement of the activation probability.

8.1.3 Action Execution

ADS-IDAC provides three primary means to generate dynamic event tree branches during the action execution phase. These branching events include timing variability, control input variability, and procedure step-skipping.

8.1.3.1 Timing Variability

Each proceduralized and non-proceduralized action executed by the operators includes timing parameters that specify the time required to perform the action. In the current version of ADS-IDAC, a three-parameter Weibull distribution is used to characterize a timing probability density function for this action time (Equation 10)

$$f(t) = \left(\frac{\beta}{\alpha}\right)\left(\frac{t}{\alpha}\right)^{\beta-1} e^{-\left(\frac{t-\mu}{\alpha}\right)^{\beta}} \qquad \textit{Equation 10}$$

Where:

- α is scale factor for the distribution (in seconds)
- β is the shape factor
- f is the probability density for time t
- t is the time to perform the associated action (in seconds)
- μ is the minimum time (in seconds)

The analyst can specify timing branching rules to generate one or more event sequence branches when the associated action is executed. If only one branch is generated, the time required to perform the action is set equal to the mean of the timing probability density function (Equation 11):

$$t_{mean} = \mu + \alpha\,\Gamma\left(1 + \frac{1}{\beta}\right) \qquad \textit{Equation 11}$$

If more than one timing branch is generated, the required action time for each branch is determined by partitioning the probability density function. For example, if *i* branches are to be generated, the timing probability density function is partitioned into *i* segments, and the mean time for each segment is calculated by Equation 12.

$$t_{mean,j} = \frac{\int_{t_{j,Lower}}^{t_{j,Upper}} t\, f(t)\, dt}{\int_{t_{j,Lower}}^{t_{j,Upper}} f(t)\, dt} = \frac{\int_{t_{j,Lower}}^{t_{j,Upper}} t\, f(t)\, dt}{\left(\frac{1}{i}\right)} \qquad \textit{Equation 12}$$

$$\textit{where } f(t) \textit{ is givien in Equation 10, and } t_{j,Lower} \textit{ and } t_{j,Upper} \textit{ found by numerically solving}:$$

$$(j-1)\left(\frac{1}{i}\right) = \int_{0}^{t_{j,Lower}} f(t)\, dt \qquad \textit{and}$$

$$\min\left(j\left(\frac{1}{i}\right), 0.999999 \right) = \int_{0}^{t_{j,Upper}} f(t)\, dt$$

In this equation, (j-1)(1/i) represents the lower cumulative probability bound for partition j while (j)(1/i) represents the upper bound. The maximum cumulative probability upper bound is limited to 0.999999 to ensure stability of the numerical solution algorithm. Calculating the mean time for each partition, rather than using a random sampling technique, allows ADS-IDAC to generate reproducible results while ensuring that the desired range of timing variability is explored.

An example application of timing branch generation is shown in Figure 26. This example shows the crew response to a steam generator tube rupture event using a procedure-following strategy. In this case, three timing branches were used to model crew timing variability in initiating the appropriate emergency operating procedure – a fast crew, a nominal crew, and a slow crew. A case with no operator actions is also shown in the figure for comparison. As seen in the figure, the generation of timing branches not only changes the time that actions are executed during an event sequence, but can also result in changes in the thermal-hydraulic response of the plant. In this case, timing variability in initiating the emergency operating procedures changes both the time that key actions are performed (such as cooldown of the reactor coolant system) and the response of key plant parameters (such as the maximum pressurizer level obtained during the accident sequence).

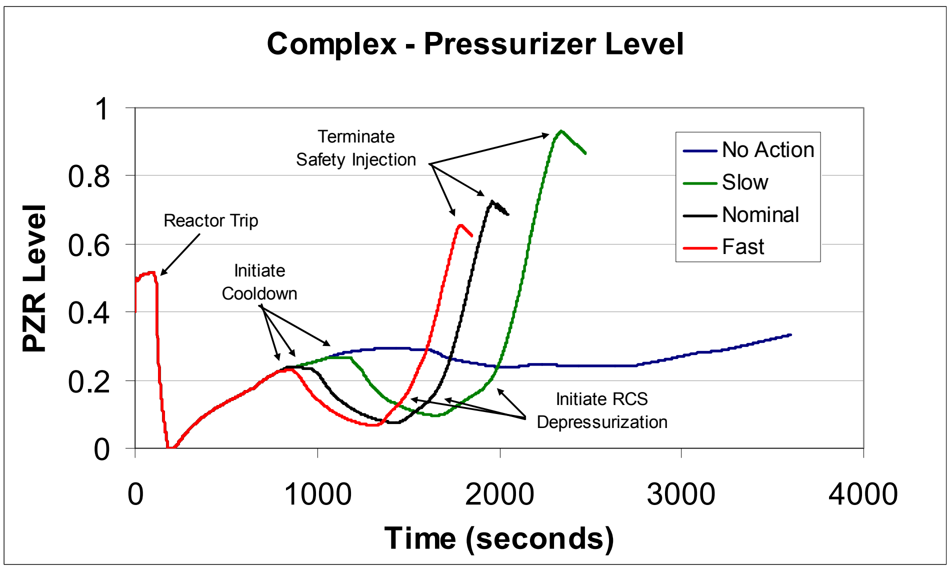


Figure 26 - Operator Timing Variability During a SGTR

8.1.3.2 Control Input Variability

When procedures require the operators to adjust a continuous variable, the crews may exhibit variability in their control input. For example, when initiating a reactor coolant system cooldown, a conservative crew may establish a relatively slow cooldown rate to ensure adequate control over a plant conditions. A more aggressive crew may establish a faster rate in order to achieve the desired end state more quickly. The analyst can model these different styles through the use of control input variability branching rules. When an action is associated with a quantitative control input (e.g., opening a throttle valve to 10% open), two or more branches can be generated to explore the effect of variations in the control input. The branching rule is characterized by a table that provides the desired control input for each branch along with a branch probability.

An example application of this branching rule is shown in Figure 27. In this scenario, the crew is using a functional recovery guideline to mitigate a complete loss of feedwater event. If no high pressure source of feedwater can be recovered, the procedure directs the operators to depressurize the steam generators in order to align a lower pressure water source (e.g., condensate pumps or fire main water). For this scenario, two control input branches were generated when the crew initiated depressurization of the steam generators. One branch modeled a conservative crew that established a relatively slow depressurization rate by opening a steam dump valve to the 5% open position. The other branch modeled a more aggressive crew that opened the steam dump valve to the 25% open position to establish a faster depressurization rate. As can be seen in Figure 27, the more aggressive crew was successful in establishing a much faster depressurization rate. Although the faster rate would be expected to reduce the time required to achieve the desired low pressure condition, in this case, the crew inadvertently activated a plant protective feature that automatically isolates the steam dump system when a high rate of depressurization is detected. Once the steam dump path is isolated, the crew will need to shift to an alternate depressurization path. Because the slower depressurization rate used by the more conservative crew did not exceed the setpoint of the automatic isolation system, they were able achieve depressurized conditions without actuating the automatic isolation plant protective feature. This example not only demonstrates

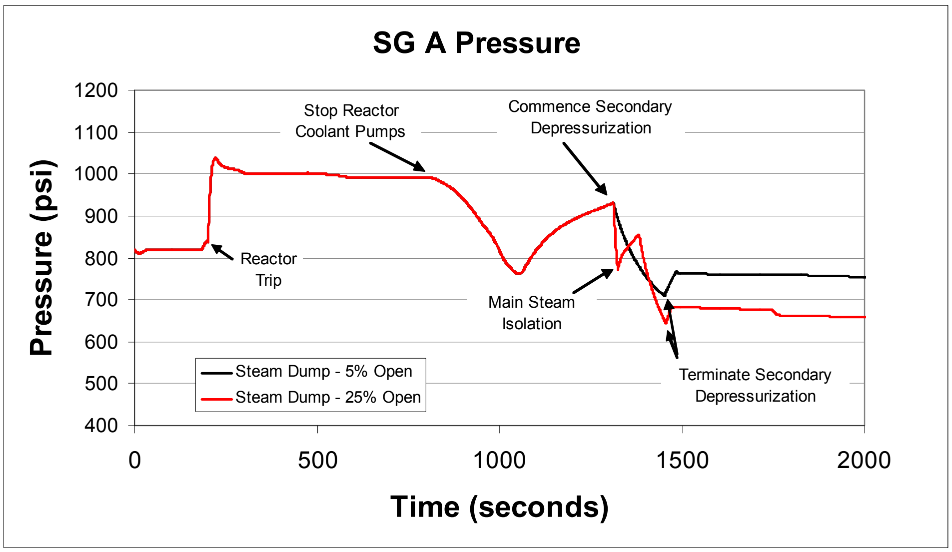


Figure 27 - Control Input Variability During a LOFW Accident

the use of the control input branching rule, but also underscores the additional level of realism that can obtained from a dynamic simulation approach to event analysis.

8.1.3.3 Procedure Following

When an operator executes either a mental or written procedure step, either the step actions or contingency actions may be omitted or skipped (see Section 6.2). The likelihood of skipping a step is dynamically calculated based on the step characteristics, the time constraint load for the operator, and the relevance of the associated action to the operator's perceived situation assessment. When a procedure step is skipped, two branches are generated: one branch where the action is performed, and another branch where the action is omitted. To prevent excessive branch generation, the analyst can specify a minimum probability threshold value for step-skipping behavior in each operator profile. If the dynamically calculated step-skipping probability is less than the minimum threshold, no branching event is generated. The minimum threshold value establishes one of the static PIFs for describing the tendency of the operator to adhere to procedural requirements (see Section 7.2.2).

8.2 Hardware Branching Events

ADS-IDAC allows the analyst to model two types of hardware failure events: (1) time dependent failures, and (2) conditional failures. Time dependent failures generate hardware faults at a prescribed time during the simulation and are used to model initiating events. Conditional failures are triggered when a specified component changes its operating state to a pre-selected target value. Time dependent failures generate only a single failure event sequence branch while conditional failures generate two event sequence branches – a success path and a failure path. Conditional failures can only be used to model failures to start (i.e., demand failures). Modeling the failure of equipment to continue to run once actuated is beyond the scope of the current research effort.

Both time dependent and conditional failures permit the operators to attempt to recover the failed equipment. If a recovery is attempted, two additional branches are generated – a successful recovery branch and a permanent failure branch. Thus, each equipment failure event can result in three outcomes: (1) the equipment does not fail, (2) the equipment initially fails but is later recovered, and (3) the equipment fails and is unrecoverable. The hardware branching rules use the Beta probability distribution to characterize both the failure and recovery probability of the equipment.

An example of a conditional failure is shown in Figure 28. In this scenario, a reactor trip is initiated at approximately 180 seconds. Upon actuation of the reactor trip, a conditional hardware failure is activated that generates two sequence branches: in one branch no additional hardware failure occurs, and in the other branch, the pressurizer power operated relief valve fails to a partially opened position.

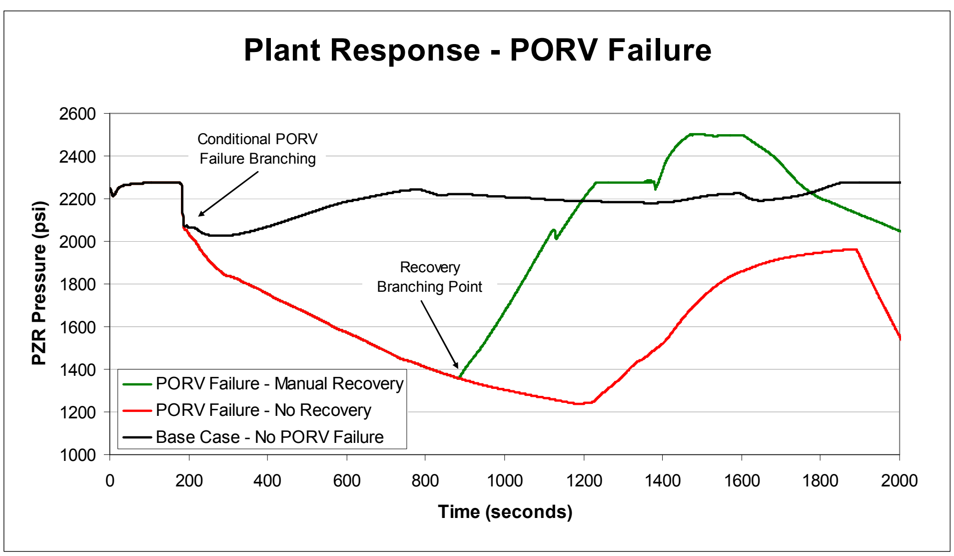


Figure 28 - Conditional Hardware Failure Branching

At approximately the 900 seconds into the PORV failure event sequence, the crew attempts to manually isolate the failed relief valve. When the recovery action is attempted, two additional branches are generated: a recovery branch where the operators re-establish normal control of the failed valve and a failure branch where the valve has permanently failed. A strength of a dynamic simulation approach is that hardware failure events can be completely integrated with the operator performance model. This integration provides rich context for each event sequence in the dynamic event tree.

8.3 Event Sequence Termination

In order to achieve complete scenario coverage using dynamic probabilistic risk assessment methods, it is necessary to explore a large number of accident sequences. However, simulating a large number of sequences often requires a significant amount of computational power and time. Excessive branch generation can lead to sequence explosion and the inability to complete an analysis in a practical amount of time. Therefore, uninteresting sequences are often terminated or truncated to permit computational resources to be focused on more interesting sequences. In general, two means to direct sequence generation are available to the analyst: specifying the number of branches generated at branching points and terminating sequences when they no longer provide useful information.

As noted in Section 8.1.3, timing and control input branching rules permit the analyst to generate two or more event sequences at each branching point. Although generating more event sequences allows better refinement in the problem solution, it can come with a high resource cost. Therefore, generation of multiple timing or control input branches should be reserved for actions that have a reasonable likelihood of leading to a degraded plant state. This topic is discussed in more detail in Sections 9.2 and 9.3.

ADS-IDAC provides several methods to terminate or truncate accident sequences. Truncation methods are based on sequence elapsed time, sequence probability, and conditional events. The following truncation methods are available in ADS-IDAC:

- Sequence Truncation Time - This option allows the user to set a maximum simulation time limit for each sequence. Once the event sequence simulation has run for the specified truncation time, the sequence is terminated.
- Sequence Probability - This option terminates a sequence when the sequence probability is less than the specified probability cutoff value. The cutoff value should carefully balance the desire to avoid excessive branch generation with ensuring that high consequence, low probability sequences are evaluated.
- Procedure Non Response Action Termination - This option allows the user to terminate a sequence if a set of procedure step expectations are not met. To implement this termination rule, a reserved keyword is used in the step non-

response action. When the ADS-IDAC simulation scheduler encounters the reserved key word during procedure execution, the sequence is terminated.

- Alarm Activation - This option terminates the sequence if an alarm with special reserved prefix is actuated. In general, this option is used to detect a significant degradation in the level of plant safety. For example, the analyst can customize alarms that actuate when excessive fuel temperatures or degraded core cooling conditions occur.

Although beyond the scope of the current research effort, sequence termination criteria can eventually be used to calculate an overall failure probability for an event of interest. For example, if sequence termination alarms were established for fuel temperatures or heat transfer conditions associated with a core damage condition, summing up the sequence probability for all sequences that were terminated by these alarms would provide an overall measure of core damage probability.

9. Implementation of ADS-IDAC Approach

As described in previous sections, implementation of the ADS-IDAC approach requires integration of a realistic nuclear plant thermal hydraulic model with a comprehensive crew behavior model. The nuclear plant model must include all plant controls, indicators, and alarms associated with significant procedural steps and skill-, rule-, and knowledge-based actions. The operator knowledge base used to drive the crew behavior model must include sufficient information to adequately represent operator knowledge, experience, and behavior characteristics. In summary, a substantial amount of information and data is required to develop an ADS-IDAC model.

In order to ensure that the operator knowledge base is capable of modeling actual crew performance, a calibration process has been developed to translate actual crew operating experience data into the ADS-IDAC knowledge base. A key consideration during the calibration process is ensuring the knowledge base includes an adequate scope to capture observed crew-to-crew variability. Only once a complete thermal hydraulic model has been built and the operator knowledge base has been populated and calibrated, can ADS-IDAC be run in a predictive mode to identify situational contexts that may lead to inappropriate crew behaviors.

This chapter describes the process used to build an adequate representation of the nuclear power plant and operator knowledge bases. In particular, a systematic approach to ensure that the nuclear plant model realistically models all system controls, alarms, and indications needed to execute significant operator actions has been developed. Similarly, a formalized process for populating the operator knowledge base is described. These calibration processes are illustrated through the use of data obtained from the OECD Halden Reactor Project during the international HRA empirical study. Finally, a systematic procedure for using ADS-IDAC in a predictive mode is described.

9.1 Thermal Hydraulic Model Development

The key mechanism used to generate plant feedback to the operator model is the nuclear plant thermal hydraulic model. The current version of ADS-IDAC utilizes the RELAP5/MOD 3.2[72] computer code to provide a transient simulation of nuclear power plant operation. The RELAP5 code can simulate a wide variety of accident initiators and provides the capability to model key safety systems, controls, and instruments. Advantages of RELAP5 include its proven capabilities as a transient analysis tool and the availability of detailed power plant models. However, due to the intrinsic limitations of the RELAP5 code, it is not currently possible to model core damage progression and severe accident phenomenology. Consequently, ADS-IDAC is currently limited to the analysis of scenarios up to the start of core damage. Adaption of ADS-IDAC to a more versatile thermal-hydraulic engine, such as the MELCOR code, has been identified as a future research activity. However, the inability of RELAP to adequately simulate core damage states is not a significant limitation to current research interests. ADS-IDAC has been developed to model the first few hours of an accident scenario – prior to the activation and staffing of the nuclear plant emergency response facilities such as the technical support center (TSC) and offsite emergency facility (EOF). Once the TSC and EOF are activated (within approximately one hour of the accident), the dynamics of crew decision making changes dramatically. Because most accident scenarios of interest would require several hours to progress to a core damage state, the inability of RELAP to capture the phenomenology of core damage progression does not impose a more stringent limitation than the existing crew decision-making models.

Although RELAP is widely used and numerous nuclear plant models have been developed to analyze accident scenarios, the level of realism required for an ADS-IDAC simulation necessitates that certain additional features be included in the plant model. These features are often not necessary when using RELAP to perform conventional deterministic analyses of accident scenarios, but are important to support crew interactions with the plant model when there are hardware failure events. Even an existing plant model will require several modifications and additions to support an ADS-IDAC dynamic PRA approach. Because existing RELAP plant models are often available, one focus of this research project has been on converting a RELAP plant model developed for a deterministic safety analysis for use with ADS-IDAC. However, the general guidelines and methods outlined here would still be applicable with a new RELAP plant model developed specifically for use with ADS-

IDAC. More detailed guidance and instructions for developing the ADS-IDAC thermal-hydraulic plant model are provided in Appendix K.

9.1.1 General Approach

Although RELAP5 plant models have been developed previously to support safety analyses and other regulatory uses, these models require some modification in order to exploit the full capabilities of ADS-IDAC. In general, ADS-IDAC requires the following modifications to an existing RELAP5 input model:

- Replacement of all conservative analysis assumptions with realistic best estimate parameters. These include trip setpoints, reactor power level, timing of automatic safety system actuations, and other key plant parameters.
- Modification to safety system models to replace simple modeling assumptions (e.g., representing a multi-train safety system with simple boundary conditions) with more realistic representations of controls, instrumentation, and alarms. These modifications include modeling of redundant trains of multi-train systems, provisions for control of significant components such as key pumps and valves, and representation of critical support systems such as water supplies and electrical power.
- Addition of systems and components that provide a significant portion of the mitigative functions provided by the abnormal and emergency operating procedures.

- Implementation of interactive control interfaces for all significant components to allow the ADS-IDAC operator model to manipulate plant components. This includes the addition of a “manual” control mode for components that normally utilize an automatic control system (e.g., feed water regulating valves and power operated relief valves).

ADS-IDAC has been successfully integrated with a three-loop, pressurized water reactor nuclear power plant RELAP model. The current plant model includes over 75 controls, 200 indicators, and approximately 100 alarms. To improve feedback to the operator, the plant model includes reactivity and core power control features such as control rod movement, boration, and turbine load adjustment. Where necessary, controls for major pumps and valves in all front line safety systems (e.g., emergency core cooling and auxiliary feed water) were also added to the existing RELAP input manual. As a result of these efforts, all major components referenced in the plant emergency procedures have been represented in the ADS-IDAC thermal-hydraulic model.

9.1.2 System and Control System Modeling

The RELAP thermal hydraulic model provides two key functions for ADS-IDAC: (1) the dynamic and realistic representation of plant parameters, and (2) the ability to interact with the thermal hydraulic model by changing component or system operating states. The ADS-IDAC control panel (see Section 3.2.1) links the operator model to the thermal hydraulic model. The ADS-IDAC environment supports a variety of operator control inputs to the thermal hydraulic model including valve position changes, starting or stopping of pumps, safety system actuations, and control system setpoint changes. In general, four possible control modes can be used for each controllable component:

1) changing the component operating mode (e.g., automatic vs. manual mode),
2) setting a specific control value for a component (e.g., throttling control valve to 50% open),
3) incrementing the control setting of a component (e.g., throttling open a control valve by an additional 10%), and
4) setting a control value based on a perceived parameter (e.g., setting the steam dump target pressure equal to the perceived main steam header pressure).

These capabilities provide sufficient flexibility to realistically model all significant operator interactions with the plant model. However, in order for the operator behavior model to control a component, a suitable system model must be added to the RELAP plant model.

Safety System Modeling

In general, only front line safety systems such as emergency core cooling and auxiliary feedwater are included in existing RELAP nuclear plant models. These systems are usually modeled as simple boundary conditions and generally lack realistic controls and status indicators. For example, the high pressure safety injection system for a three loop pressurized water reactor is usually modeled using only three time dependent junctions to provide injection flow to each reactor coolant loop (Figure 29). The time dependent volume generally represents an infinite source of fluid, while the time dependent junction provides a specific mass flow rate of injection fluid based on a predefined formula (e.g., mass flow might be a function of reactor coolant system pressure). While these abstract modeling elements can accurately represent the main safety system functions under certain conditions, they lack a direct correspondence to the controls and indicators found in a nuclear plant control room. In order to improve the realism of the model, it is necessary to modify the system models to add multiple trains, control valves, and piping elements.

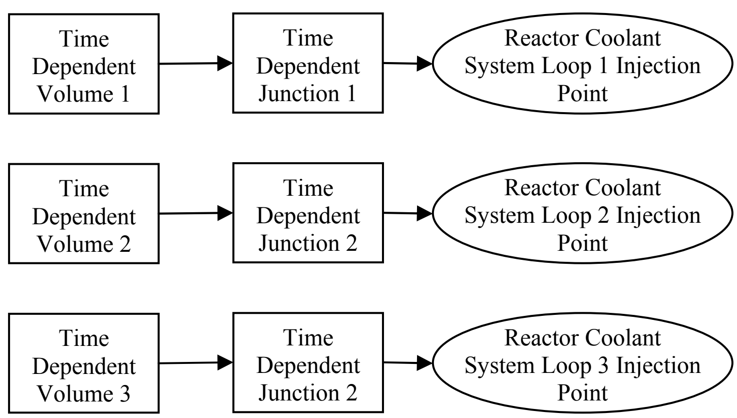


Figure 29 - Typical RELAP Mitigative System Model

The modified system model for use with ADS-IDAC includes separate subsystem trains for safety system and realistically models potentially hydraulic dependencies for loop injection flow (Figure 30). In this modified model, time dependent volume "A" and time dependent junction "A" represent one pump train of a two train system. Each pump train joins into a common header (as is usually the case for an actual plant safety system) and then splits to provide water flow to each reactor coolant system loop. The benefits of the modified system model include:

- Capability to independently control multiple safety system trains and individual loop flows;
- More realistic representation of actual pump head/flow characteristics since the improved model more closely matches the actual plant configuration;
- Improved modeling of injection flow dependencies between the reactor coolant loops. Since the injection flow is supplied from a common header (the common hydraulic volume), the improved model does not decouple the loop injection to one loop from the other loops (e.g., high injection flow to loop 1 will decrease injection flow to loops 2 and 3).

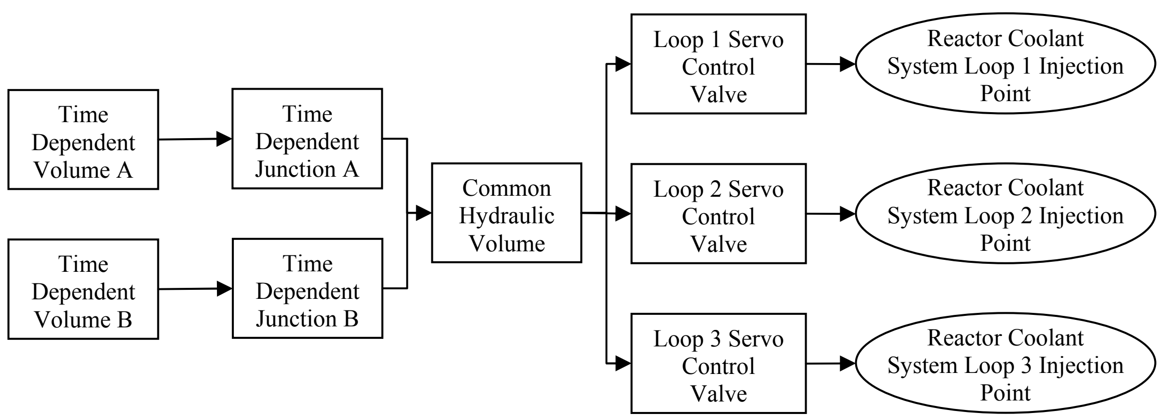


Figure 30 - Mitigative System Modeling in ADS-IDAC

The system level modeling for all front line safety systems in an existing RELAP input deck (e.g., high pressure safety injection, low pressure safety injection, and auxiliary feedwater) must be reviewed to adjust overly simplified modeling approaches and ensure that the model closely approximates the actual configuration of the system.

In general, RELAP input deck plant models developed for generalized safety analyses do not include detailed models for support systems such as electrical power, cooling water, or lubricating oil. These support system dependencies can either be modeled directly or through use of appropriate logical dependencies within the plant model. When the added realism of a detailed model is not required, it is usually sufficient to use a simple logical dependency. For example, a loss of offsite electrical power can be modeled by establishing logic conditions that cause the failure of all components powered directly from offsite power (e.g., reactor coolant pumps, turbine-generator auxiliary systems, and condenser circulating water). A similar procedure can be used for other support systems such as cooling water and lubricating oil. In this manner, support system dependencies are simulated without the need for developing a detailed support system model.

Non-Safety System Modeling

Nuclear plant systems are generally divided into two categories: safety-related systems and non-safety-related systems. Safety related systems include all the plant systems, structures, and components needed to establish licensing requirements for the protection of public health and safety. Although non-safety systems may be used to accomplish similar functions as safety systems, they are generally not credited in licensing safety evaluations (unless their operation can result in an adverse impact). Safety systems are subject to stringent quality assurance requirements and are typically referenced by the emergency operating procedures. Non-safety systems are generally associated with power generation and include major plant systems such as the turbine and generator, and portions of the main steam and feed systems.

Because the RELAP thermal hydraulic program is generally used to perform accident analyses in support of nuclear power plant licensing, non-safety related components, equipment, and systems normally used during power operation are often omitted or simplistically modeled. For example, if the turbine-generator system is included in the plant model, turbine load is often set at a full power value with no means available for adjustment. Similarly, existing RELAP models for non-safety control systems such as makeup and letdown functions do not provide a sufficient range of flexibility to model an adequate range of operator interactions. Therefore, the RELAP plant model must be reviewed to identify features that need to be added to provide the capability to model operator behaviors that might occur during power operation. Typical examples include:

- turbine generator load control;
- non-safety related interlocks or protective features (e.g., turbine runbacks);
- charging/letdown system operation;
- nuclear reactivity control systems (e.g., control rods, emergency boration); and
- condenser steam dump control.

The addition of these features provides control over functions frequently referenced in the plant operating procedures, and allows the operator model to interact with the systems normally used to control plant conditions. Because use of safety systems may preclude continued operation or delay a return to power operation, operators may initially use non-safety systems to maintain plant conditions after an abnormal event. In particular, these systems often support troubleshooting actions by the control room crew.

Control Systems

An existing RELAP plant model will usually include several automatic control systems to maintain key thermal hydraulic parameters at a predefined setpoint. Typical examples include pressurizer level control, reactor coolant pressure control, feed water control, and steam pressure control systems. Often, these RELAP control systems correspond to actual control systems in the nuclear plant. Occasionally, a control system will be added to simulate operator interactions with the model. In addition to improving the realism of the plant model, control systems enhance the stability of the thermal hydraulic model. For example, in the absence of a dynamic control system, small errors in initializing the thermal hydraulic model (such as a mismatch between steam and feed water flow rates) can cause significant deviations over time.

In order to ensure control systems are appropriately modeled for use with ADS-IDAC, all automatic systems in the RELAP model are reviewed to determine if they correspond to an actual control system or if they were added to simulate operator actions. Since the crew behavior model in ADS-IDAC controls all operator interactions with the plant model, any control system model that does not correspond to an actual plant system must be disabled or removed. Additionally, any automatic control system that does correspond to an actual plant system must be modified within the RELAP input deck to permit the ability to manipulate equipment from the ADS-IDAC environment. In general, nuclear plant automatic control systems allow the operators to take manual control of the final actuating device(s) or change the control set point. In addition to allowing more realistic plant control, the manual control mode can also be used to mitigate certain hardware failures and initiating events. Ideally, three modes of operation should be provided for each automatic control system:

i. fully automatic - the automatic control system positions the final actuating device(s) to maintain target parameter at the nominal setpoint;
ii. fully manual control - the operator positions the final actuating device(s); and
iii. setpoint control - the control system automatically positions the final actuating device(s) to maintain the targeted parameter at the operator selected setpoint.

These operational modes provide sufficient flexibility to carry out most control system manipulations referenced in plant procedures.

9.1.3 Plant Model Verification

To the extent possible, the ADS-IDAC plant thermal hydraulic model was compared to actual plant operating characteristics to verify key modeling assumptions. Output from the ADS-IDAC plant model used for this research project was compared to that from the Halden Reactor Project FRESH simulator. In addition, plant data from the Updated Final Safety Analysis Report (UFSAR[9]) for the reference ADS-IDAC plant model was used to ensure the accuracy of the simulation model. Both the ADS-IDAC plant model and FRESH simulator are based on a three reactor coolant loop pressurized water reactor design and use very similar control and protection systems. Several key plant operating parameters are compared in Table 12. As can be seen in Table 12, the ADS-IDAC plant model compares favorably with both the reference plant UFSAR information and the FRESH simulator.

There are several differences between the ADS-IDAC model and the HAMMLAB FRESH simulator. Notably, the full power pressurizer water level and wide range steam generator levels are not consistent. This difference could be due to either physical differences in the respective plant models (e.g., component configuration differences) or instrumentation calibration differences. Additionally, the full power level of the ADS-IDAC plant model appears to be approximately 2% higher than the FRESH simulator (based on full power main steam flow). During the calibration phase of this research study (described in Section 9.2.2), it was also noted that there were a number of other key plant differences between the ADS-IDAC and FRESH plant models. Notably, the base ADS-IDAC model includes main steam system non-return valves which are intended to prevent backflow from one steam generator to another steam generator during certain types of main steam system ruptures. The FRESH simulator model does not appear to include this

[9] Title 10, Part 50.71, "Maintenance of records, making of reports" of the Code of Federal Regulations requires each nuclear power plant licensee to periodically update the Final Safety Analysis Report (FSAR) for the facility. The UFSAR provides a description of plant systems, operating characteristics, and the plant response to design basis accidents.

Table 12 - Plant Model Comparison

Parameter	ADS-IDAC Model	Reference Plant Final Safety Analysis Report	HAMMLAB FRESH Simulator
Reactor Power	2660 MW	2660 MW	-
Average RCS Temperature	577 F	577 F	577 F
Core Exit Temperature	-	-	624
Hot Leg Temperature	610 F	610 F	-
Core Differential Temperature	66 F	68 F	-
Narrow Range SG Water Level (100% Power)	44%	-	44%
Wide Range SG Water Level (100% Power)	60%	-	72%
SG Pressure (100% Power)	805 psig	-	813 psi
Total Main Steam Flow Rate (100% Power)	3230 lbm/sec		3150 lbm/sec
Normal Reactor Coolant System Operating Pressure	2260 psig	2235 psig	2235 psi
Pressurizer Water Level (100% Power)	51%	-	56%
Pressurizer PORV Relief Capability	60 lbm/sec @ 2200 psia	55 lbm/sec @ 2235 psi	(Note 1)
SG Atmospheric Dump Valve Relief Capability	75 lbm/sec @ 800 psi	38 lbm/sec @ 1035 psi	~85 lbm/sec @ 800 psi
Total AFW Flowrate	175 lbm/sec	~180 lbm/sec	185 lbm/sec
Motor Driven AFW Pump Flowrate (at nominal SG Pressure)	~44 lbm/sec	350 gpm (~45 lbm/sec)	-
Turbine Driven AFW Pump Flowrate (at nominal SG Pressure)	~87.5 lbm/sec	700 gpm (~90 lbm/sec)	-
High Head Safety Injection Pump Flowrate (at normal operating RCS pressure)	~55 lbm/sec @ 2000 psi	150 gpm/pump (~40 lbm/sec)	~54 lbm/sec @ 2000 psi

Note 1: Although the PORV relief capacity could not be determined from available information, it was concluded that the FRESH reactor coolant system depressurization rate for a partially opened pressurizer PORV was consistent with the ADS-IDAC model. This implies that the PORV capacities for the ADS-IDAC model and the FRESH simulator are similar.

plant-specific feature. Also, the logic and setpoints used to actuate some of plant engineered safety features are different in the plant models. In particular, the FRESH model includes an engineered safety actuation when the pressure differential between steam lines exceeds a specific threshold; the ADS-IDAC base model does not include this feature. These model differences represent a potential limitation on the applicability of ADS-IDAC results to HAMMLAB operator experiments. For scenarios where the contextual background is sensitive to these design differences, the ADS-IDAC model may not adequately represent plant and crew behavior. The impact of these plant model differences is discussed in more detail in Section 10.2.

9.2 Calibration Mode

Even when subject to similar personnel selection, training, and administrative requirements, nuclear plant control room operators can exhibit significant crew-to-crew variability during non-routine events. A strength of the ADS-IDAC simulation approach is the ability to systematically model the sources of variability. Therefore, a focus of the ADS-IDAC calibration process is to identify and model these potential sources of crew-to-crew variability. In general, ADS-IDAC captures crew variability within three main categories: procedure execution, operator mental models of reactor plant functions, and operator preferences.

In general, the mapping of actual crew behaviors to ADS-IDAC branching rules follows the four step process shown in Figure 31.

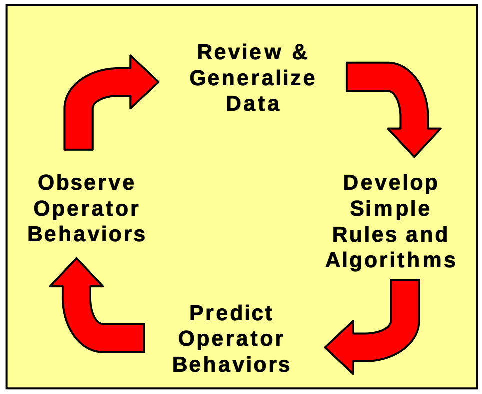


Figure 31 - Branching Rule Mapping Process

The first step in the mapping process is to gather operator data from either control room observations or simulator exercises. The operator data is then reviewed during the data abstraction and generalization step to identify commonalities and differences among crews. Although this step requires expert judgment, the goal is to group observed crew-to-crew variations into the smallest number of behavioral categories that are capable of capturing significant crew differences. For example, if variabilities are discovered in the time required to perform a certain action, binning crews into slow, nominal, and fast categories might provide a sufficient representation of the crew variability. In the next step, simple branching rules and algorithms are developed to map observed behaviors to appropriate branching rules. The final step involves the prediction of operator behaviors using these branching rules and comparing the predictions to the observed behaviors. The branching rules can then be modified in an iterative process to provide an adequate level of agreement between the observed crews and the ADS-IDAC simulation. The goal of this process is to characterize crew behavior in a manner that preserves generalizability without the development of overly prescriptive branching rules.

9.2.1 General Calibration Approach

The objective of the ADS-IDAC general calibration procedure was to develop a model of operator cognitive processes capable of replicating the plant response obtained during actual nuclear plant control room crew simulator exercises. For this research study, quantitative simulator log data from the HAMMLAB FRESH simulator was available to support the calibration process. The simulator log was obtained during experiments conducted in the Fall of 2006 and provides a time history of key plant parameters, operator control inputs, alarms, and simulator control commands. This data source provided an extremely detailed record of plant status that can be readily compared to the output from the ADS-IDAC RELAP thermal-hydraulic model. However, qualitative data characterizing the crew decision-making processes and the operators' underlying motivation and mental state were not available. Therefore, the cognitive decision-making functions carried out by the crew were inferred from the crew interactions with the reactor plant. For example, a crew that shut down the reactor early in an accident scenario was judged to have a lower tolerance to engage in troubleshooting activities or to maintain the plant at power with degraded conditions. A consequence of this data limitation is that the calibration process focused on developing a reasonable crew decision-making model that could of replicate observed plant responses to specific accidents, rather than explicitly reproducing actual crew decisions, goals, and motivations. While the ADS-IDAC control room crew operator model can be reasonably well calibrated to generate a manual reactor trip or other control input at a time or plant condition consistent with observed data, it is impossible to calibrate ADS-IDAC to each operator's detailed cognitive processes without additional data regarding the specifics of these processes. In other words, the ADS-IDAC model currently can be calibrated to capture the *results* of the cognitive decision-making process (e.g., modeling an operator action to manually initiate a safety system), but does not address the calibration of specific details of the cognitive *process* (e.g., when the decision is made to carry out an action to manually initiate a safety system and the basis for this decision). This approach is consistent with the IDAC model characterization of human error defined in terms of plant needs rather than specific operator behaviors. While the calibration process is believed to generate a plausible and reasonable representation of crew cognitive processes, a detailed validation of the crew cognitive model was beyond the scope of this study. If qualitative data adequately describing the crew cognitive decision-making process were to become available, incorporation of this information into the ADS-IDAC model could be readily accomplished.

As previously described, the calibration process includes four general steps: collection and analysis of actual crew performance data; generalization of observed data; formulation of appropriate branching rules; and prediction of crew behaviors using the selected rules. The process is iterative so that if the selected set of branching rules does not adequately replicate the observed plant response, the branching rules are adjusted until adequate ADS-IDAC performance is obtained.

Collection and Analysis of Actual Plant Data

The first step in the calibration process is collecting, organizing, and analyzing plant data collected from studies of actual control room crews. Ideally, data from multiple crews for the same scenario should be collected to facilitate identification of areas where performance exhibits a high degree of crew-to-crew variability. For this research project, the main data source is simulator log data from the Halden Reactor Project FRESH simulator. This data consists of a time history of all key plant parameters, control room alarms, and operator control inputs for fourteen separate crews. The plant data is organized in a format to aid the identification of key operator inputs and areas of crew-to-crew variability through the use of a spreadsheet that provides a time history of key plant parameters for each crew. A review of the governing plant procedures is then conducted to map key proceduralized actions to observed changes in plant parameters. For example, a procedure step that directs the operators to shut a main steam isolation valve for a steam generator can be mapped to operation of the associated control switch or a rapid decrease in the steam flow from the affected steam generator. The goal is to map as many unambiguous proceduralized actions as possible to the observed plant data in order to establish a series of performance benchmarks during the accident scenario. The relative timing of these benchmarks can then be used to establish the pacing each crew used during scenario. Additionally, differences in the rate of change of key plant parameters between crews during the scenario indicate variability in control inputs. For example, differences in steam flow following execution of a procedure step to initiate a reactor coolant system cooldown may indicate differences in steam dump valve control inputs for the associated crews.

Once a set of benchmarks is established between the procedural requirements and actual plant performance, the data is reviewed to identify the following:

- Crew timing variations – The relative spacing of the benchmark points for each crew can be directly translated into timing variability.
- Variations in control inputs – Some crews may use aggressive control inputs that result in rapid changes in plant parameters while other crews are may exhibit more conservatism when changing plant conditions.
- Thresholds for Alternate Actions - Procedures may give the crew multiple options to accomplish the same function. In these cases, procedures will typically specify a preferred method to accomplish a function, but allow the crew to utilize an alternate method if the preferred method is unavailable or ineffective. For example, reactor coolant system pressure can be decreased by either opening the pressurizer spray valve or the power operated relief valve. Although a procedure may express a preference of one method, the decision when to switch from the primary method to an alternate method may be left to the crew's judgment. In these cases, it may be useful to analyze the rate of change in the target parameter or holistically assess plant status to gain insights as to when the crew may decide to abandon a primary method. Obviously, it would be better to obtain this insight directly from the crew when available, rather than inferring it from plant data.

- Skipped Steps – In some cases, a plant conditions may not map to expected proceduralized actions if a crew intentionally or unintentionally skips the associated procedure step(s).
- Deviations from Procedures – Some changes in plant parameters will not map to proceduralized actions. In these cases, the crews may have performed a knowledge-based action outside the scope of the plant procedure.

Once the mapping between the governing procedures and plant data is complete and specific areas of variability in the observed data from different crews have been identified, the operator behaviors are grouped and generalized.

Development of Operational Narratives and Generalization of Observed Behaviors

The behaviors that are catalogued during the data collection and analysis phase are reviewed to identify commonalities and differences among the crews. In order to capture the widest range of crew-to-crew variability with the smallest set of branching rules, it is useful to develop an operational narrative to explain crew behaviors during a scenario. These operational narratives can be used to collapse several observed deviations and complex behaviors into a single crew decision or preference. The narratives can also be used to explain dependencies between a series of actions taken by a crew.

The development of these operational narratives is similar to what is done with other cognitively-based HRA models. For example, the Méthode d'Evaluation de la Réalisation des Missions Opérateurs pour la Sûreté (MERMOS) HRA method relies on the development of "little stories" that describe how a human failure event occurs [90]. Similarly, A Technique for Human Event Analysis (ATHEANA), one of the HRA methods developed by the US NRC, includes the development of a base scenario description and associated deviations [10]. In fact, the guidance from the MERMOS or ATHEANA approaches can be used to assist in the development of the ADS-IDAC operational narratives. Once an adequate set of operational narratives capable of representing observed crew behaviors is developed, these narratives are translated into a set of branching rules.

Formation of Appropriate Branching Rules

The goal of this phase of the calibration process is to translate the insights developed during the formulation of the operational narratives into specific branching rules. As discussed in Section 8, branching rules have been developed to capture performance variability and a wide range of crew preferences and tendencies. Table 13 provides a summary of mapping of typical sources of crew-to-crew variability to the appropriate ADS-IDAC branching rules.

Verification of Selected Branching Rules and Adjustment of Model

Once an initial set of branching rules is developed, the ADS-IDAC simulation code is executed to determine the adequacy of calibration. Plant data obtained from the ADS-IDAC simulation is compared to the actual observed crew behaviors to determine if the branching rules replicated observed thermal-hydraulic plant

Table 13 - Mapping Crew Variability to ADS-IDAC Branching Rules

Type of Variability	Source of Crew-to-Crew Variability	ADS-IDAC Branching Rule(s)
Procedure Execution	Selection of component target control settings (e.g., throttle valve positions, manual controller set points)	Action control value branching rule
	Time required to execute procedure actions	Action time branching rule
	Recovery of failed equipment	Equipment failure and recovery branching rule
	Omission or skipping of procedure steps	Procedure step-skipping branching rule
	Interpretation of potentially ambiguous procedure criteria (e.g.,, translation of qualitative descriptors such as "decreasing", "increasing", or steady" into context-specific quantitative criteria)	No branching rule currently available. Conditional simulation runs can be run to examine the use of different quantitative thresholds for qualitative criteria.
Operator Mental Models	Training, experience, and diagnostic capabilities	Mental belief branch probability branching rule. The operator knowledge base should include a sufficient spectrum of mental beliefs to capture crew variabilities due to training and knowledge.
	Time required to initiate automatic memorized actions	Mental procedure activation time branching rule
Operator Preferences	Selection of high level goals and problem solving strategies	Troubleshooting probability and procedure use probability branching rules
	Tendency to rely on memorized information rather than control panel readings	Use of memorized information branching rule
	Delays or breaks during procedure execution (e.g., crew briefings)	Action time branching rule. Appropriate procedure hold steps must be incorporated into ADS-IDAC procedures to model delays or breaks.
	Tendency to manually perform anticipatory safety actions prior to automatic actions	Mental belief branch probability rule. The operator knowledge base must include appropriate mental belief(s) to activate a memorized mental procedure that implements the anticipatory manual action.

performance. As noted previously, the goal of this calibration process is to realistically replicate the plant performance observed during actual control room scenarios, not necessarily capture the details and timing of specific crew decisions, goals, and motivations. However, to the extent that more detailed qualitative operator cognitive data is available, it can be used to further refine the decision-making model.

9.2.2 Model Calibration Using HAMMLAB Data

The Halden Human-Machine Laboratory (HAMMLAB), a research facility affiliated with the international Organization of Economic Cooperation and Development (OECD), recently participated in a human reliability analysis empirical investigation intended to improve the understanding of the strengths and weaknesses of various HRA methods [91, 92]. As part of this study, fourteen certified nuclear plant control room crews performed four simulator scenarios in the FRESH pressurized water reactor (PWR) simulator at HAMMLAB to collect empirical human performance data. Two of the scenarios were associated with a steam generator tube rupture event and two of the scenarios were associated with a loss of feedwater event. During the scenarios, the control room crews utilized HAMMLAB specific emergency operating procedures which were adapted from the operating crews' home plant procedures. The crew performance data included simulator log data which provided a detailed summary history of key parameter values, alarms, and control manipulations. The data from the steam generator tube rupture scenarios was used to calibrate ADS-IDAC using the process outlined in Section 9.2.1. The goal of this effort was to assess the feasibility of modeling actual control room crews with simple branching rules in ADS-IDAC. The data from the loss of feedwater scenarios was used to test the predictive capabilities of ADS-IDAC – these scenarios are further described in Section 10.2.

A steam generator tube rupture (SGTR) is caused by a break in one or more of the heat exchanger tubes within a steam generator. The SGTR break diverts coolant water from the reactor core cooling system to the secondary cooling system and the main turbine. Because a SGTR causes a loss of reactor coolant, operators would normally observe symptoms of a loss of coolant accident including decreasing pressure and reduced pressurizer level. The diversion of radiologically contaminated coolant to the steam generators can result in an increase in steam generator water inventory and various radiation alarms in the normally uncontaminated secondary coolant system. Two variations were used for the SGTR scenarios: a relatively uncomplicated base case, and a more complex case that included additional plant hardware failures. The base case scenario was a relatively straightforward scenario without additional plant equipment failures, and the emergency procedures provided adequate crew guidance. Conversely, the complex scenario included a number of additional failures such as a main steam line break, failure of radiation monitoring equipment, and failure of the pressurizer PORV. Only the base case calibration was completed for this research project, as the predictive capabilities of ADS-IDAC had not yet been confirmed. The remainder of this section describes calibration activities using base case SGTR scenario data.

The main mitigative actions for a SGTR specified in the HAMMLAB procedures include: identification and isolation of the ruptured steam generator (SG), cooldown of the reactor coolant system (RCS), depressurization of the reactor coolant system to stop leakage from the RCS into the ruptured SG, and termination of safety injection to prevent overfill of the RCS. In order to assess the pace and timing of crew actions, the emergency procedures were reviewed to identify operator actions associated with these objectives that could be unambiguously identified in the simulator log data. These actions serve as timing benchmarks and permit crew-to-crew comparisons of procedure execution speed. The simulator data was also reviewed to identify other crew behavior differences such as interpretation of qualitative procedure criteria (e.g., interpretation of “stable”, “increasing”, “decreasing”), crew actions not covered in emergency procedures (e.g., initial power reduction), and actions that were not consistent with procedural requirements. As a result of these efforts, the following areas of crew-to-crew variability were identified for the base case scenario: procedure pace and timing; initial power reduction; initiation of reactor trip; early auxiliary feedwater isolation; RCS cooldown rate; early termination of cooldown and depressurization; pressure equalization; and RCS depressurization method.

- Procedure Pace and Timing – As noted earlier, the SG tube rupture emergency procedure (emergency procedure E-3) contains three main objectives: (1) isolate the ruptured steam generator, (2) cool down the reactor coolant system to establish an adequate subcooling margin, and (3) depressurize the reactor coolant system to terminate the leakage through the failed steam generator tube. Each of these action objectives is associated with a series of closely related operator activities; each set of objective actions is known as an action block. Based on the selected timing benchmarks, three procedure hold points were created in ADS-IDAC to adjust the pace of procedure execution. The procedure holds capture crew briefings, unexpected delays, and the general procedure step execution speed of the crew. Procedure hold points were located as follows (see Figure 32):

i. Briefing Hold #1 - Prior to the start of the SGTR emergency procedure, Halden Procedure E-3. The length of this hold point was based on the time delay between the actuation of the emergency core cooling system and closure of the main steam isolation valve on the ruptured SG in Step 3 of Halden emergency procedure E-3.

ii. Briefing Hold #2 - Prior to the reactor coolant system cooldown initiated in Halden Procedure E-3, Step 7. The length of this hold point was based on the time delay between closure of the main steam isolation in Step 3 of Halden Procedure E-3 and initiation of steam dumping in Step 7.

iii. Briefing Hold #3 - Prior to reactor coolant system depressurization initiated in Halden Procedure E-3, Step 16. The length of this hold point was based on the time delay between the termination of the reactor coolant system cooldown in Step 7 and the initiation of depressurization in Step 16 of Halden Procedure E-3.

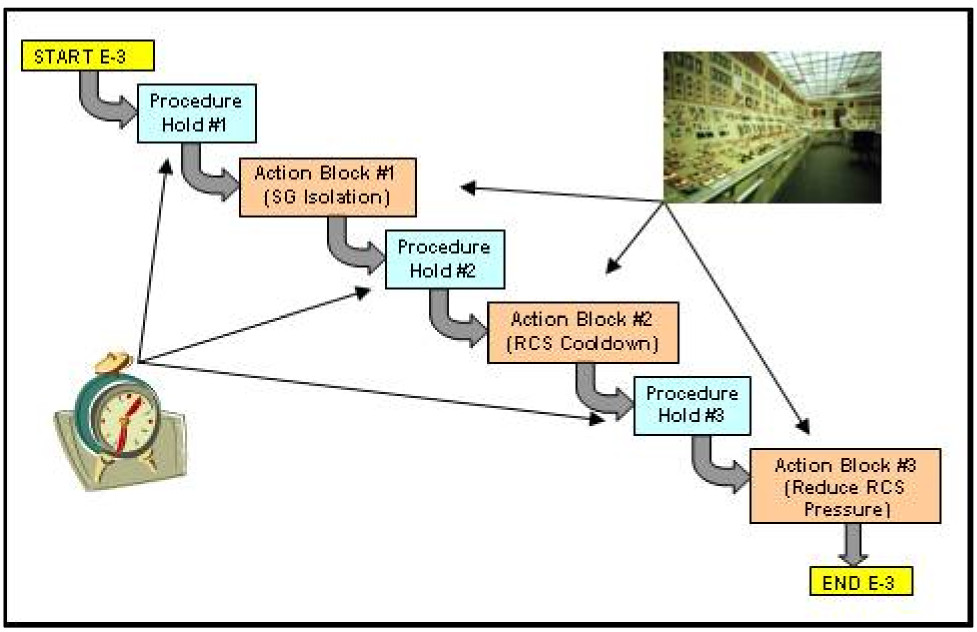


Figure 32 - Modeling Timing Variability with Action Blocks

In addition to the above procedurally driven action blocks, a review of the HAMMLAB data indicated that operators may have isolated auxiliary feedwater flow to the ruptured steam generator prior to being directed to do so in the Halden emergency operating procedure E-3[10]. Actual crew timing data obtained from the FRESH simulator log files were fit to three-parameter Weibull distributions to obtain a probabilistic distribution for each procedure hold point (see Table 14).

Table 14 - SGTR Crew Timing Probability Distributions

Scenario	Parameter	Procedure Hold #1	Procedure Hold #2	Procedure Hold #3	Isolate AFW
Base Scenario	Minimum time (μ)	375	25	100	230
	Scaling Parameter (α)	382	212	148	331
	Shape Factor (β)	0.992	1.38	1.07	1.47
	Kolmogorov-Smirnov Test Statistic (K-S)	0.292	0.148	0.166	0.122
	Critical K-S Value (0.05 significance)	0.349	0.349	0.349	0.349

These timing probability distributions can then partitioned into two or more branch events to represent variations in crew pacing. For example, Figure 33 illustrates the generation of five timing branches based on experimental data from the HAMMLAB facility for procedure hold #2 during the base scenario. The crew performance data is fitted to a three parameter Weibull distribution, which is then used to calculate the mean time over each discrete timing interval. The

[10] Halden Procedure E-3, Step 4 directs the operators to isolate auxiliary feedwater flow to the ruptured steam generator when the narrow range level is above 10% (the SG level requirement is intended to ensure that the ruptured tube is covered with water to provide additional radionuclide scrubbing for the leaking reactor coolant). Halden Procedure E-0, Step 12 (which the operators would perform prior to E-3, Step 4,) contains similar guidance but does not specify a minimum SG water level requirement. A review of the HAMMLAB data indicated that seven of the fourteen crews for the base case (and one of fourteen crews in the complex case) isolated auxiliary feedwater prior to achieving the 10% minimum level specificed in E-3. This indicates that when the occurrence of a SG tube rupture is relatively easy to diagnose, operators may have performed this action prior to reaching step 4 of procedure E-3. Consequently, within ADS-IDAC, isolation of auxiliary feedwater was modeled with a skill-based rule that was activated when the operators perceive a ruptured steam generator. The time delay for the execution of this action was calibrated to the actual time delay between the initiation of the steam generator tube rupture event and the crew action to isolate auxiliary feedwater flow.

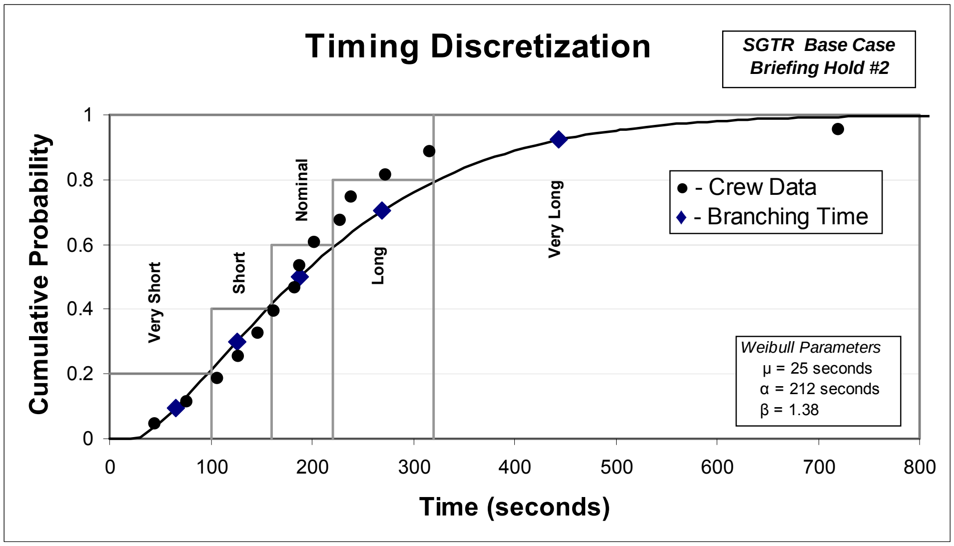


Figure 33 - ADS-IDAC Branching Times

interval size is determined by evenly spacing the desired number of branches over the cumulative probability distribution. In this case, five intervals were selected, each representing a cumulative probability increment of 0.2.

- Initial Power Reduction – Several crews delayed actuation of the emergency core cooling system and initially attempted to reduce reactor power. To capture this behavior, a mental belief was created that would trigger a turbine load reduction when the crew perceived the initial indications of a reactor coolant system leak and implemented the "troubleshooting" goal.

- Initiation of Reactor Trip – The crews that did not initiate a power reduction typically tripped the reactor relatively early in the accident scenario (generally within two to three minutes of initiation of the SGTR). This variance can be modeled in ADS-IDAC by activation of the maintaining global safety margin goal. The ADS-IDAC decision-making model may activate this goal when an abnormal condition is detected and the troubleshooting goal is disabled or bypassed (see Section 5.2). When the global safety margin goal is active, the operators will immediately initiate the emergency operating procedures, which will lead to a manual reactor trip. In order to establish the timing of manual reactor trip initiation, it is necessary to adjust the abnormal condition detection threshold, one of the static PIFs that characterize operator preferences and redundancies (see Section 7.2.2). A lower abnormal detection threshold will decrease the time between SGTR initiation and manual trip because the operator requires less information to conclude that an abnormal condition has developed. A sensitivity study was conducted on the impact of changing the abnormal signal detection threshold. As can be seen in Figure 34, the time to initiate a manual reactor trip following initiation of a SGTR increases as the abnormal condition detection threshold is increased. At a value of 0.7, the manual reactor trip is delayed long enough to allow the automatic reactor trip (due to low reactor coolant system pressure) to shut down the reactor plant. The relationship between the abnormal

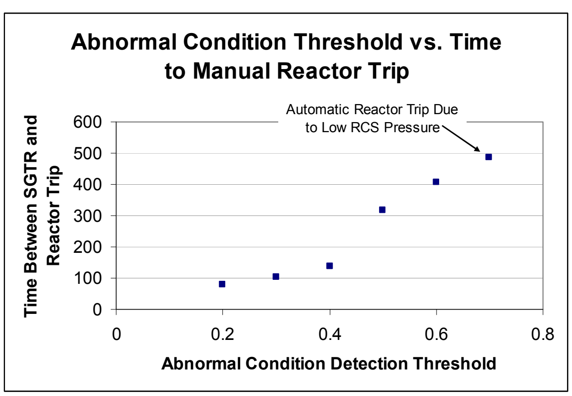


Figure 34 - Effect of Signal Detection Threshold

detection threshold and the time to initiate a manual trip are consistent with the cutoff reinforcement model discussed in Section 2.5.1.1. Specifically, a higher threshold value could be used to model an operator who has accumulated substantial "near-miss" experience and has tendencies toward risk-seeking behavior.

- Early Auxiliary Feedwater Isolation – The HAMMLAB procedures permitted the crew to isolate AFW flow to the ruptured SG early if an SGTR event was indicated. To capture observed crew timing variability for this action, the actual time lag between the availability of indications of an SGTR and AFW isolation was fit to a Weibull probability distribution (see Table 14). Mental procedure activation time branches were generated to partition this probabilistic distribution to model crew timing variability.

- RCS Cooldown Rate – Although the emergency procedures directed the crews to dump steam at the maximum rate, an excessive cooldown rate can actuate an automatic main steam isolation signal that isolates the steam dump path. A main steam isolation requires the crew to shift the cooldown method from the condenser steam dump to the SG atmospheric steam dump valves and delays the cooldown. Crew variability in selecting a steam dump rate was modeled by generating action control value branches to control the throttle position of the steam dump valve (a more open throttle position corresponds to a faster cooldown rate).

- Early Termination of Cooldown and Depressurization – Several crews were unable to reach the target temperature or pressure during the RCS cooldown and depressurization steps. These conditions can be modeled in ADS-IDAC with conditional runs that utilize overly conservative target values for temperature and pressure.

- Pressure Equalization – The HAMMLAB procedures eventually direct the crew to equalize pressure between the RCS and the ruptured SG in order to terminate the tube leakage. It was noted that several crews were unable to obtain a stable equalized pressure condition. This variability was modeled by activating a step-skipping branch that bypassed RCS/SG pressure equalization.

- RCS Depressurization Method – The HAMMLAB procedures required the operators to initiate the RCS depressurization using normal pressurizer spray, but allowed the pressurizer PORV to be used if the crew believed that the depressurization was too slow. It was noted that crews used different interpretations of what constituted a "slow" depressurization. This behavior was modeled by adding a procedural expectation to the ADS-IDAC model that prompted use of the PORV when the depressurization rate dropped below a threshold value. Crew variability was captured by performing conditional simulation runs using different threshold values for the transition depressurization rate.

As an example application of capturing observed crew-to-crew variability using ADS-IDAC, a simulation was run with the goal of replicating the performance of two specific crews. For this example, five branching events were sufficient to model the differences between the two selected crews for the Base case scenario (Figure 35).

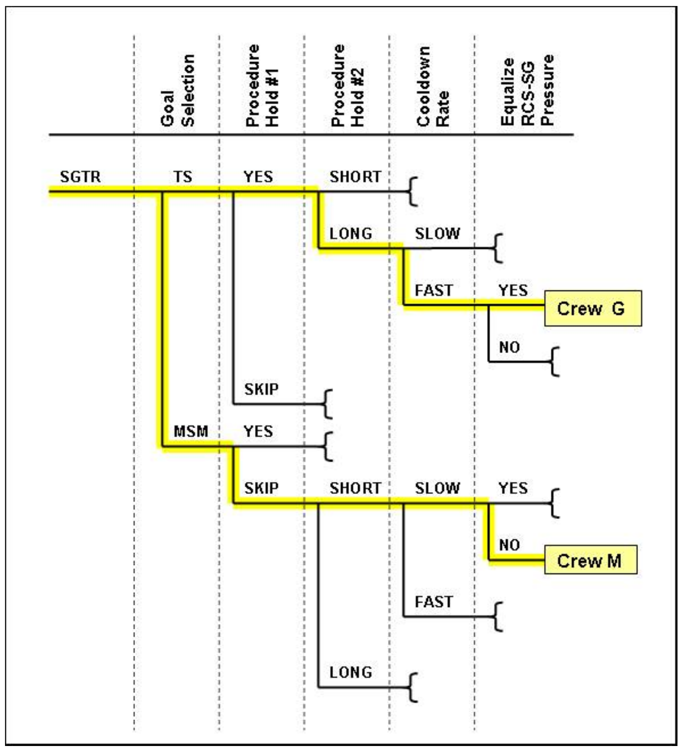


Figure 35 - Example SGTR Branching Events for HAMMLAB Crews

These branching events modeled the following crew-to-crew variabilities:

- Goal Selection – Crew G attempted to initially reduce power and delayed the manual reactor trip and actuation of the emergency core cooling system until approximately seven minutes after the start of the SGTR. Conversely, Crew M manually tripped the reactor approximately two minutes after the start of the event. This behavior variance was modeled by enabling the troubleshooting goal for Crew G, and blocking the goal for Crew M.

- Procedure Timing – Crew M moved relatively quickly through the emergency procedures and initiated reactor coolant system cooldown relatively faster than Crew G. This timing variability was modeled by Crew M skipping a crew brief prior to initiating Procedure E-3 (Procedure Hold #1) and moving relatively quickly through the remainder of the procedure (Procedure Hold #2). Crew G was modeled with a nominal length crew briefing prior to the start of procedure E-3 and moved at a relatively slow pace through the procedure (Procedure Hold #2).
- Control Input Variability – The SGTR emergency procedure requires the crews to rapidly cool down the reactor coolant system by dumping steam at the "maximum" rate. The interpretation of "maximum" in this context does not refer to the overall capability of the plant (i.e., the maximum steam dumping rate for the system), but rather includes a consideration of other plant constraints. For example, a steam dump rate that is too fast will cause an automatic main steam isolation due to the rapid decrease in steam pressure. This isolation will interrupt the plant cooldown as the crew to shifts from the normal condenser steam dump system (which is now isolated) to use of the SG atmospheric relief valves. Therefore, dumping steam at too fast a rate will likely result in a longer time to cooldown as the crew must recover from the automatic main steam valve closure. In this scenario, Crew M dumped steam at a relatively slow rate that resulted in a steady RCS cooldown without

actuation of the automatic isolation. Crew G attempted a relatively rapid cooldown that resulted in isolation of the steam dump system. This variability was modeled with a control value branching rule.

- RCS-SG Pressure Equalization – One of the final steps in the emergency procedure is to operate the pressure control system to equalize the pressure differential between the reactor coolant system and the ruptured steam generator. Once the pressure differential is reduced, the leakage of reactor coolant into the ruptured steam generator is terminated. Failure to equalize pressure will result in continued leakage into the steam generator. During this scenario, Crew G successfully equalized RCS and SG pressure, while Crew M failed to reduce the pressure differential. The failure to achieve an equalized pressure condition results in several adverse impacts, including a sustained high ruptured SG pressure leading to release of coolant through the atmospheric steam dump and a continued loss of reactor coolant.

A comparison between reactor coolant system pressure for the actual crews and the ADS-IDAC simulation are shown in Figure 36 and Figure 37.

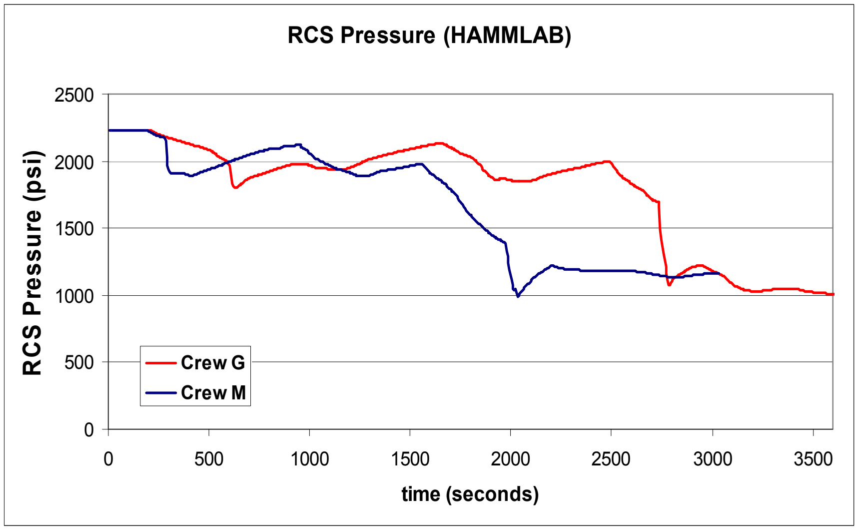


Figure 36 - Example HAMMLAB SGTR Scenario Plant Data

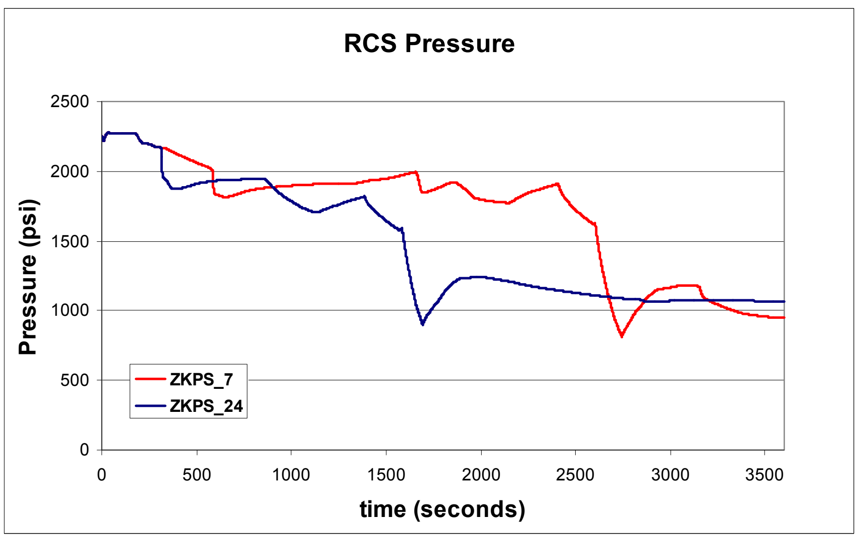


Figure 37 - ADS-IDAC Calibration to HAMMLAB Data

As can be seen, ADS-IDAC effectively captures significant differences between these crews, including performance of a downpower maneuver by Crew G, a faster procedure execution pacing by Crew M, and the pressure equalization between the RCS and ruptured SG performed by Crew G. Thus, using relatively few branching events, ADS-IDAC effectively captured the crew-to-crew variability of actual control room operators.

9.2.3 Calibration Insights and Conclusions

The ADS-IDAC calibration process provides an effective means to realistically represent the sources and consequences of observed crew-to-crew variabilities. Complex operator mental models of plant behavior that guide crew actions can be represented within the ADS-IDAC mental belief framework. Branching rules can be created to simulate slow or fast procedure execution speed, skipping of procedure steps, reliance on memorized information, activation of mental beliefs, variations in control inputs, and equipment failures. More importantly, this project has demonstrated that a realistic spectrum of control room crew-to-crew differences can be captured with a relatively small number of branching rules. This approach allows the generalization of observed data within the IDAC framework. Because model based HRA techniques such as ADS-IDAC attempt to capture underlying cognitive processes that drive crew behaviors, these models provide an efficient framework for capturing actual operator performance data such as timing of operator actions, mental models, and decision-making activities.

9.3 Predictive Mode

Once the ADS-IDAC model is appropriately calibrated, it can be used to explore situational contexts that may lead to inappropriate operator actions. A major goal of this research project was the development of an analysis tool capable of predicting when knowledge-based errors of commission may occur, such as those which may arise is when knowledge-based actions are incorrectly used because a crew fails to develop an accurate situational assessment. In addition to supporting the analysis of information driven errors, ADS-IDAC is also capable of exploring the impact of certain types of errors of omission (e.g., skipping procedure steps) and the consequences of crew-to-crew performance variabilities (e.g., timing and control input variability). A key strength of the simulation approach is that the consequences of inappropriate actions can be explicitly determined by the thermal-hydraulic model. Therefore, human actions that lead to degraded plant states (e.g., excessive fuel temperatures or compromised core cooling) can be identified.

A general procedure has been developed for using ADS-IDAC to predict situations where human error events may occur. This predictive approach, described in Section 9.3.1, is a qualitative tool at the present time. Although ADS-IDAC is currently capable of quantifying the probability of each accident sequence, further research is needed to ensure that probabilistic calculations are adequately calibrated. Despite this limitation, ADS-IDAC is capable of providing significant insights regarding possible crew responses to accident scenarios and the impact of these responses on the plant state. To demonstrate the capabilities of ADS-IDAC, the predictive analysis procedure was exercised using the HALDEN empirical study loss of feedwater scenario. The results of the comparison between ADS-IDAC predictions and actual crew behavior are provided in Section 10.2.

9.3.1 General Analysis Approach

As described in Section 8, ADS-IDAC can explore a range of potential accident scenarios by activating branching rules when certain conditions are met. Each branching point includes two or more individual event branches which represent distinct combinations of system and operator states. Collectively, the branching points describe the topology of a discrete dynamic event tree (DDET) associated with an initiating event. A specific accident sequence is defined by the unique path through the DDET branching points from the initiating event to an end state. Accident sequences that lead to degraded end states (e.g., excessive fuel temperatures or otherwise degraded core cooling conditions) can be reviewed to determine if inappropriate human actions contributed to the adverse plant state. Because ADS-IDAC is capable of representing knowledge-driven actions not specified in plant procedures, this approach shows promise for identifying situational contexts that lead to errors of commission.

The generation of branching points is controlled by a set of general rules that define the specific activation conditions for each branching point. Although the branching rules are predefined, the creation of specific branching points during an accident sequence depends on the dynamic behavior of the reactor plant and operator decision-making models. Consequently, a simulation approach is needed to determine the branching points that are generated along a specific accident sequence trajectory. This project has demonstrated that complex variations in crew-to-crew performance can be captured with a relatively small set of generalized branching rules. In order to identify the branching rules that are appropriate for a specific accident analysis, a general analysis procedure has been developed. This procedure involves a review of procedural requirements; operator knowledge, skills, and experience; and crew preferences and tendencies. Once sufficient background information has been collected and analyzed, this information is translated into a set of branching rules for the accident scenario of interest.

Review and Analyze Procedural Framework

The relevant operating procedures provide a starting framework for identifying potential sources of crew-to-crew variability. Such variations may include differences in procedure pace and timing, control input variations, and the perceived importance of each procedure step. In order to identify these procedurally driven factors, the following methodological approach was developed:

1) Identify major control inputs and component actuations (e.g., stopping or starting of major components, opening or closing significant flow control valves). Actions associated with the same goal (e.g., establishing a cool down or depressurization of the reactor coolant system) are often clustered together within a procedure and can delineate major phases in the emergency

procedures. Crews encountering these blocks of related actions during a procedure may elect to hold a briefing prior to executing significant plant actions. These blocks are also convenient locations to insert procedure delays to model crew variations in the pace and timing of procedure execution. For example, depending on the leadership style of the senior reactor operator, the crew's familiarity with the procedures, and overall operator experience, crews might have significant variability in their procedure execution speed. Both timing delays and the conduct of briefings and meetings can be addressed by the inclusion of procedure hold points with action time branching rules. Branching options can also be exercised to simulate fast, slow, or nominal crews (see Section 8.1.3.1).

2) Identify possible operator control input variations. Procedurally ambiguous control manipulations where the operators may use too little or too much control input (e.g., opening valve more or less than desired) are potential areas for crew-to-crew variance. Variations in control inputs can be modeled with appropriate control value branching events (see Section 8.1.3.2).

3) Identify ambiguous threshold criteria in procedure steps by searching for key words such as "increasing," "steady," "decreasing," and "uncontrolled". These keywords highlight steps where crews may exhibit variability in interpreting procedural criteria. For example, crews may use different threshold criteria for judging acceptability. Further, interpretation of criteria

may depend on the situational context and the crew's knowledge and experience. For example, a crew might conclude that a decreasing trend in pressure actually indicates a stable condition if the cause of the decrease is well understood and a direct result of operator actions. Crew variability associated with interpretation of potentially ambiguous criteria can be modeled with several conditional simulation runs to explore a range of threshold values.

4) Assess the main objective of each procedure step. While some procedure steps may support the main procedure objectives, other steps may be associated with supporting activities that may have a lower salience or priority for the operators. Therefore, steps that are more directly linked to the main procedure objectives would be expected to have a lower likelihood of being skipped. However, some steps that do not appear to be directly linked to the procedure objectives may improve the efficiency of the procedure, establish prerequisite conditions that support later actions, or delay the onset of core damage. Examples of such actions include:

    a. Tripping the reactor coolant pumps following a loss of secondary heat sink. This action reduces heat input to the reactor coolant system but does not directly support the main procedure objective to recover feedwater flow to the steam generators. Consequently, the action has a safety impact (reduced time available for recovery) but may not be salient to the operators.

    b. Blocking automatic safety injection actuation signals prior to an intentional cooldown or depressurization to prevent an unnecessary safety system actuation. The failure to block an actuation signal under these circumstances can unnecessarily delay mitigative actions, but may not have a high degree of relevance to the operators during the accident since it does not directly provide a mitigative function

5) Adjust operator knowledge base and procedure step profiles. The operator knowledge base includes a system decomposition that maps each indicator, component, and alarm to the functions they support (see Section 4.3.1). Additionally, each procedure step includes profiling information that describes the objectives and complexity of the step actions (see Section 4.3.3). The operator knowledge base and procedure step profiles can be adjusted to reflect the expected level of crew knowledge and experience with the procedure.

6) Identify all steps that require that require continuous monitoring of a plant parameter by the operator. These steps often require the operator to initiate a plant transient (e.g., opening a pressurizer PORV), closely monitor a target parameter (e.g., RCS pressure), and terminate the transient when a target value is reached. If the operator fails to monitor the target parameter at regular intervals, initiation of the required actions to terminate the associated transient could be delayed. By activating the use of memorized information or adjusting the operator's information gathering capability (see Section 7.2.1), these types of information processing limitations can be modeled.

7) Identify continuous action steps. These steps require the operators to take specified actions when a monitored parameter exceeds a defined threshold at anytime during procedure execution. This is contrasted with the continuous monitoring discussed in the previous paragraph since continuous action steps

are not normally activated as a result of deliberate action by the operator to initiate a plant transient. Therefore, the conditions requiring activation of a continuous action step may be less salient to the operator. Continuous actions can take several forms – steps included in the procedure "fold out" page that apply during the entire procedure, notes and cautions within the procedure, and special instructions that direct the operators to take certain compensatory measures only during specific parts of the procedure. For example, the recovery procedure for an uncomplicated reactor trip might direct the operators to initiate emergency core cooling if pressurizer level decreases below 10%. Because these types of steps require the operators to periodically monitor certain plant parameters and perform activities outside the normal flow of the procedure, one expects them to have a higher likelihood of being missed or implemented late. These steps can be generally modeled with appropriate mental beliefs that include activation of the parent procedure and the associated parameter threshold values as prerequisite conditions (see Section 5.4). Variability in the execution of continuous action steps can either be modeled through the selection of an appropriate activation probability or through the use of information filtering or biasing.

8) Identify procedure steps that refer to components that are assumed to have failed during the accident scenario. These procedure steps may trigger the operator to initiate recovery or compensatory actions.

Extend Operator Knowledge Base

The knowledge base represents the memorized information, skills, and abilities available to the operator. Specific items included in the knowledge representation include written procedures, diagnostic guidance, a functional decomposition model of reactor plant systems, and rules governing the activation mental beliefs. All anticipated procedures associated with the scenario of interest should be included in the operator knowledge base, along with additional controls, indicators, and alarms needed to support the major procedural actions. When new components are added to the ADS-IDAC control panel, they must be mapped to their associated functions in the operator's system decomposition. Depending on the level of crew experience, training, and abilities, adjustments can be made to the diagnosis matrix (see Section 5.5.1.3) and the system decomposition map.

A key part of the operator knowledge base is the set of mental beliefs that represent discrete observations and decisions. Collectively, mental beliefs characterize each operator's situational assessment of the plant state. Therefore, mental beliefs that may be pertinent to the scenario of interest should be identified based on operator training and experience. For example, mental beliefs could include "uncontrolled steam generator level increase" for a steam generator tube rupture scenario and "faulted steam generator" for a main steam line break. In addition, non-proceduralized actions that may extend the time available until a safety system must be actuated should be identified. Typical examples include maximizing reactor coolant system makeup during a primary system leak and reducing steam loads during a partial loss of feedwater. Also, the conditions under which operators might manually activate safety systems or place automatic control systems in a manual mode should be identified. Each of these non-proceduralized actions can be modeled with skill- or rule-based procedures that are activated by mental beliefs. Crew variability can arise from the failure to reach appropriate belief states, delays in acting on beliefs, and setting a higher or lower evidence threshold for a belief. Each of these factors can be adjusted to represent crew-to-crew situational assessment variability (see Section 5.4).

Characterize Crew Capabilities, Preferences and Tendencies

Each individual operator in ADS-IDAC is provided with profiling data that guide his or her behavior. The operator profile also includes data needed to: (1) calculate performance influencing factors; (2) define the operator's tendencies to skip procedure steps or pursue specific problem solving strategies; (3) manage memorized information; and (4) establish the timing of actions and communications. The flexibility afforded by the operator profile allows the simulation of a variety of operator performance tendencies. Specifically, performance influencing factors associated with problem solving styles, perception and appraisal of information, and utilization of memorized information can all be captured within the operator profile. In order to ensure that the operator profile accurately models the crews of interest, a review of plant organizational factors, operator training, and operating experience should be conducted to assess the following factors:

- the tendency of crews to rely on previously memorized information rather than use of recent information obtained directly from the control panel indicators. The use of memorized information can reduce activity execution time but may result in the use of outdated and incorrect information.
- the procedural adherence tendency of the crews. Procedural steps in ADS-IDAC have three main components: (1) initial action activity, (2) expectations associated with the initial action activity, and (3) a non-response action that is executed if the action expectations are not met. The operator may skip either the initial action activity or the non-response action (evaluation of the action expectations cannot be skipped in the current ADS-IDAC model).
- the crew's threshold level for concluding that an accident has occurred. Based on accumulated evidence, the ADS-IDAC diagnosis engine calculates a value representing the potential for the observed information being related to an accident condition. When this calculated value exceeds a preset threshold, the crew will transition to the emergency operating procedures and shut down the reactor.
- the tendency to rely on knowledge-based troubleshooting rather than written procedures. Operating crews may attempt to address an emergency condition through the use of actions based on their knowledge and experience rather than through written procedures. The ADS-IDAC knowledge base allows the analyst to specify the crew's tendency to use knowledge-based reasoning approaches to problem solving rather than written procedures.

Once each of these crew preferences and tendencies is reviewed, appropriate operator profiling factors can be identified. In general, these profiling factors are specified for each member of the crew by static PIFs (see Section 7.2).

In addition to identifying crew preferences and tendencies, the dynamic PIFs include several static constants that specify each operator's capabilities for processing information, assessing plant status, and handling time stress. The appropriate dynamic PIF tuning factors should be adjusted to appropriately model these crew performance attributes (see Section 7.3).

Identify Specific Branching Rules for the Analysis

Once sources of crew-to-crew are variability are identified, appropriate branching rules must specified for the analysis. Branching rules cover three broad categories: hardware failure events, operator mental models, and crew preferences and tendencies. Specific guidance for creating branching rules can be found in Section 8 and Appendix K. In order to minimize the potential for sequence explosion and ensure efficient use computational resources, appropriate sequence end states and truncation thresholds should be identified. End states should be based on a minimum set of critical parameters and associated thresholds. For example, if core damage is the degraded state of interest, a sequence can be terminated when fuel temperatures exceed a high threshold value (e.g., 2200° F), when the heat transfer is severely degraded, or when a substantial portion of the core is uncovered. Sequences can also be truncated when they represent low probability events, or if the sequence exceeds a specified time limit. Care should be exercised when setting probability truncation values in order to ensure that low probability, high consequence scenarios are not unnecessarily excluded from consideration. Time truncation limits should reflect inherent limitations in the ADS-IDAC decision-making model. In particular, the ADS-IDAC decision-making model is not valid once decision-making authority is shared with emergency operations facilities. This generally limits the time frame for the ADS-IDAC analysis to the first hour or two of an accident scenario.

In order to organize ADS-IDAC simulation runs and facilitate data analysis, a simulation matrix should be developed that specifies the detailed computer simulations that will be performed. Due to the computer processing limitations, it is not currently practicable to run a single ADS-IDAC simulation capable of exercising all branching points. The main difficulty is the exponential increase in accident sequences as the number of branches increases. As the number of sequences increases, the time to complete a simulation run can become prohibitive. One method of overcoming this difficulty is through the performance of conditional runs where a reduced number of branching rules are activated and all other branching rules are suppressed. If the combination of branching rules is selected with care, the analyst should be able to capture a wide range of potential crew behaviors with minimal processing effort. In general, the analyst should select a combination of three or four branching events that reflect risk significant scenarios. The simulation matrix should ensure that all significant branching rules are explored.

9.3.2 Human Error Prediction

As discussed in Section 2.3.3, human error within the IDAC model is defined in terms of the human operator failing to meet the needs of the nuclear plant system. Because the nuclear plant consequences of operator actions are directly simulated within ADS-IDAC, actions that lead to an unsafe plant state can be considered to be in error. Compared to traditional static risk assessment methods, ADS-IDAC can provide a more realistic assessment of human error events by directly determining the effect of operator behaviors on plant thermal hydraulic parameters. This shifts the analysis from an assessment of isolated operator actions to a more holistic assessment of the control room situational context and the integrated impact of a spectrum of possible operator actions on the reactor plant

An important consideration when assessing the potential consequences of human actions in the nuclear control room environment is identifying the sources of behavior variabilities among operating crews. As previously noted, even when personnel selection, training programs, and administrative programs are consistently implemented, nuclear plant operators can exhibit significant crew-to-crew performance variabilities. These performance variabilities can arise from differences in crew knowledge, skills, and experience; crew specific organizational factors; and operator preferences and tendencies. Variability is normally defined in comparison to a normative case. For the purposes of this study, normative crew behavior is defined as: (1) the execution of procedural requirements without deviation and in a manner consistent with safety analysis assumptions; and (2) the execution of non-proceduralized actions in a manner consistent with operator training and the recognized and accepted skill-of-the-craft. Within this context, non-normative behaviors can be associated with either beneficial or undesirable operator actions. An underlying assumption of the ADS-IDAC approach is that deviations from the normative (or expected) set of operator behaviors following an accident can sometimes lead to a degraded plant state. By examining the situational context associated with undesirable deviations, the factors leading to human error events can be better understood.

10. Model Validation

In order to verify, to the extent possible, that the ADS-IDAC is capable of appropriately simulating operator behavior, this research project includes a validation component, focusing on validating the results and outcomes of the model rather than its functional details. Typical validation measures include face validity, content validity, and criterion related validity [21, 93]. Face validity refers to the degree to which the model captures important and relevant behaviors as judged by potential users. Although face validity is a subjective and relatively weak measure of overall validity, it can influence the level of confidence that users have in model results. Content-related validity refers to the inclusion of all pertinent factors that can influence the capability of the model to meet its objectives. In this case, content-related validity implies that the ADS-IDAC model addresses factors that significantly influence the identification of nuclear plant error forcing situations. Criterion-related validity refers to the capability of the model to predict operator behaviors and, where possible, provide results consist with other modeling techniques. This validation effort addresses the following specific elements:

- Does the model appear to provide reasonable results for a range of accident scenarios? (face validity);
- Is the model capable of simulating real-world operator behaviors associated with significant historical events such as the Three Mile Island accident? (face validity);

- Does ADS-IDAC include pertinent human behavior elements? (content validity);
- Is the model consistent with other modeling techniques? (content validity);
- Does the ADS-IDAC adequately predict operator behavior? (criterion validity).

The validation effort was structured to determine both the reasonableness and predictive power of the model. Specifically, a spectrum of accident scenarios was analyzed to determine if the model provides realistic representations of normative crew performance. A general approach for modeling sources of crew-to-crew variabilities for postulated accident scenarios was then applied to the loss of feedwater (LOFW) HRA empirical study scenarios in order to predict potential human error events. Finally, a comparison of the ADS-IDAC predictions to the actual empirical study results was made. Insights regarding re-calibration of the dynamic model are provided. Due to the lack of available human performance statistical data for nuclear plant operators [94], the validity assessment focused on qualitative factors, rather than a rigorous quantitative statistical approach. As more nuclear plant operator human performance data becomes available, it should be possible to revisit the quantitative aspects of ADS-IDAC validation as part of a future validation effort. For example, ongoing U.S. Nuclear Regulatory Commission efforts to collect human performance data under the Human Event Repository and Analysis (HERA) [95] may eventually provide sufficient quantifiable data to support additional validation of ADS-IDAC. Despite the limitations associated with the limited availability of human performance data, the current validation effort has resulted in significant improvements in the capabilities of the ADS-IDAC model and demonstrated methods for capturing the rich data obtained from empirical simulator-based studies.

10.1 Initial Validation (Content Validity)

The first stage in the ADS-IDAC validation effort was to ensure that the ADS-IDAC simulation model included the pertinent features necessary to predict human error. This involved a qualitative comparison between the key features of the IDA cognitive model, other human error prediction techniques such as ATHEANA and SPAR-H, and the specific ADS-IDAC implementation model. It should be noted that the content-related validity assessment used a qualitative approach. Specifically, the focus of this review was whether or not the ADS-IDAC model included the key features needed to fulfill its intended purpose of predicting human error events.

The first step of the content validity review was to compare the model elements included in ADS-IDAC to the formal IDA model. For each phase of the IDA cognitive process, the key attributes of the IDA model were compared to their implementation within the ADS-IDAC computer code. The results of this comparison are provided in Table 15. Based on this comparison, it is clear that the ADS-IDAC simulation model does include the key features and elements described in the IDAC cognitive model.

Table 15 - Comparison of IDA Model to ADS-IDAC Implementation

Cognitive Phase	Key Attributes of the IDAC Model [96]	Implementation in ADS-IDAC
Information Processing	Information Filtering (attention, biasing, capacity)	The control panel scanning module includes factors to account for memory capacity and relevance of the information. The filtering module includes the capability to model quantitative biasing of parameter values. Section 4.1.2 and 4.2.
	Information Grouping	The knowledge base functional decomposition provides a means to group information that relates to a common functional element. Section 4.3.1.
	Memorization and Retrieval	The operator memory model includes factors to model the use of memorized information and the information retention capability of the operator. Section 7.2.1.
	Intentional and Non-Intentional Information Gathering	ADS-IDAC models both actively and passively collected information. Section 4.1.
	Prioritization of Information	The ADS-IDAC control panel scanning queue includes consideration of the priority of the information. Section 4.1.2.
Decision Making	Problem Solving High Level Goals and Subgoals	ADS-IDAC models three of the four IDAC high level goals. The goal of "maintain equipment safety margin" is addressed by mental beliefs and skill- or rule-based actions (see Section 5.4). In addition, an intermediate "Monitoring" goal has been added to the model. Section 5.2.
	Problem Solving Strategies	ADS-IDAC currently models six of the nine IDAC problem solving strategies (the instinctive response and direct matching strategy have been combined). The limiting reasoning, ask for advice, and trail and error strategies are not currently modeled. Section 5.3.
	Goal and Strategy Selection Rules	Although the specific rules used for goal and strategy selection differ from the IDAC model, a tractable process is used for goal and strategy selection. Sections 5.2 and 5.3.
Action Execution	Slip errors	The procedure step-skipping module is capable of modeling slip-type errors. Section 6.2.

In most human reliability assessment methods, a set of performance influencing shaping factors is used to condition the probability of human error based on a variety of factors. These factors typically include consideration of procedure adequacy, stress, operator knowledge and experience, ergonomics, and environmental factors. In order to ensure that the ADS-IDAC model is consistent in its treatment of these factors, the performance influencing factors currently modeled in ADS-IDAC were compared to those described in the IDAC model [20] and the SPAR-H HRA method [88]. Although SPAR-H is not a cognitively-based human reliability assessment method, it is widely used by the U.S. Nuclear Regulatory Commission for events assessment and the accident sequence precursor program[11]. Table 16 provides the results of this comparison. As can be seen, the current version of ADS-IDAC includes many of the influencing factors described by both the IDAC and SPAR-H methods. There are still several areas where the ADS-IDAC model can be further improved and extended. For example, ADS-IDAC does not currently include consideration of environmental and physical factors. Further, the treatment of organizational and team related factors could be enhanced to improve the accuracy and realism of the model. However, these additional model improvements are beyond the scope of the current research effort.

In addition to the comparisons with cognitive modeling elements and performance influencing factors, the ADS-IDAC model includes a number of other features that serve to increase confidence in the model results. These include the contents of the operator knowledge base, the basic model for activation of skill- and rule-based actions, and the calibration of the thermal-hydraulic plant model. In particular, ADS-IDAC includes the following validating features:

[11] The NRC's accident sequence precursor program systematically analyzes significant operational events at commercial power plants to identify potential precursors to severe core damage accident scenarios. This proram is used to trend nuclear power plant risk and provide feedback to regulatory programs.

- The operator knowledge base includes an accurate representation of plant emergency procedures. In addition to the procedure model being capable of representing both the structure and content of plant procedures,
  the procedure model is based on the HALDEN emergency operating procedures for the FRESH simulator. These procedures provide a reasonable approximation to the actual procedures used in nuclear power plants of similar design. As discussed in Section 9.1.3, the ADS-IDAC plant model compares favorably to the FRESH simulator. Skill- and rule-based operator action tendencies, such as those described for the ATHEANA HRA method [85], can be easily adapted for use within ADS-IDAC using the mental belief model.
- The ADS-IDAC mental belief model (Sections 4.3.2 and 5.4) is based on the Recognition Primed Decision (RPD) making model, which describes the decision making process utilized by experienced persons during dynamic conditions [14]. The main features of the RPD model comport favorably with the nuclear plant control room environment, notably dynamic conditions, action-feedback loops, high stakes, and experienced decision makers working in a team environment. The RPD approach has also been extended to other similar high time stress decision-making environments such as air traffic control modeling [97].
- The three-loop pressurized water reactor model currently implemented in ADS-IDAC includes all significant controls, alarms, and indicators needed to execute key mitigative actions described in the HALDEN emergency procedures. Additionally, as described in Section 9.1.3, the ADS-IDAC

nuclear plant model compares favorably to both the reference plant design and the HALDEN FRESH simulator.

Table 16 - Performance Influencing Factor Comparison

IDA Performance Influencing Factor Group	IDAC Model [20]	SPAR-H [88]	Implementation in ADS-IDAC
Cognitive Modes and Tendencies	• Alertness • Attention to Current Task • Attention to Surrounding Environment • Bias	-	Situational Assessment Module (Section 5.5)
Emotional Arousal	• Stress • Frustration • Conflict • Pressure • Uncertainty	Stress/Stressors	Not explicitly modeled, but dynamic time-constraint load and information load PIFs provide a surrogate measure of stress.
Strains and Feelings	• Time-Constraint Load • Task-Related Load • Non-Task Related Load • Passive Information Load • Confidence in Performance	Available Time	Dynamic PIFs for time-constraint and information loading (Section 7.3.1 and 7.3.2) are used to model strains and feelings.
Perception and Appraisal	• Perceived Severity of Consequences Associated with Current Diagnosis • Perceived Criticality of System Condition • Perceived Familiarity with Situation • Perceived System Confirmatory or Contradictory Response • Perception of Alarms • Perceived Decision Responsibility • Perceived Complexity of Strategy • Perceived task Complexity • Perception of Problem Solving Resources • Awareness of Role, Responsibility	Complexity	The dynamic PIF factor for criticality of system condition (Section 7.3.3) provides a measure of the perceived severity of the situation. Complexity factors can be included in the procedure model (Section 4.3.3). Perception of alarms and the associated follow-up actions can be handled through the use of control panel scanning (Section 4.1.2) and the activation of alarm response procedures via mental beliefs (Section 5.4).
Memorized Information	• Knowledge, Experience • Skills • Memory of Recent	Experience, Training	Knowledge and experience is modeled within the operator knowledge base (Section 4.3).

IDA Performance Influencing Factor Group	IDAC Model [20]	SPAR-H [88]	Implementation in ADS-IDAC
	Diagnosis, Actions, and Results • Memory of Incoming Information		Operator skills can be addressed through the use of appropriate mental beliefs and associated skill- or rule-based procedures (Section 5.4). Operators can memorize and retrieve information (Section 7.2.1)
Intrinsic Characteristics	• Self Confidence • Problem Solving Style • Morale, Motivation, Attitude	-	Problem solving style can be directly addressed by selection of appropriate crew preferences and tendencies as established by static PIFs (Section 7.2)
Environmental Factors	• Harsh Environment • Physical Access • Visual and Audio Effects of Surroundings	-	Not currently modeled in ADS-IDAC
Conditioning Events	• Hardware • Software • Human • Others	-	Although conditioning events are not explicitly modeled, the operators can retain memories of previous information and events (Section 4.1.1).
Organizational Factors	• Work Process • Human-System Interaction • Safety and Quality Culture • Work Environment • Tool Availability • Tool Adequacy • Procedure Availability • Procedure Adequacy and Quality	Procedures Ergonomics Work Processes	ADS-IDAC is capable of accurately representing both the content and structure of plant procedures (Section 4.3.3). Certain elements of safety culture can be addressed by appropriate activation threshold with static PIFs (Section 7.2.2).
Team-Related Factors	• Cohesiveness • Coordination • Communication Availability • Communications Quality • Composition • Leadership	-	ADS-IDAC models a control room crew consisting of a senior reactor operator (decision-maker) and a reactor operator (action-taker) (Sections 3.1.3 and 7.4). Although communication quality is not directly monitored, the time to communicate can be addressed with static PIFs (Section 7.4).
Physical Factors	• Fatigue • Physical Limitations	Fitness for Duty	Not currently modeled in ADS-IDAC

These factors collectively serve to increase the confidence that the ADS-IDAC model includes key modeling elements to support a reasonable evaluation of the factors that may lead to human error events.

10.2 Model Behavior and Response (Face Validity)

The objective of the face validity assessment is to check that the ADS-IDAC model provides "expected" results a variety of accident scenarios. For example, following an uncomplicated reactor shutdown, the plant interactions arising from the ADS-IDAC operator model should bring the plant to a safe and stable condition in a short period of time. Furthermore, the model should be able to cope with increasingly complex scenarios while still maintaining the ability to reproduce normative crew behavior. For this study, two accident scenarios were evaluated – the first scenario involved a reactor trip event while the second investigated the model response to conditions similar to those that occurred during the Three Mile Island accident in 1979.

10.2.1 Reactor Trip Response

The objective of this phase of the study was to evaluate the ADS-IDAC model response to a reactor trip event. Reactor trips are a relatively common initiating event and are classified as anticipated operational occurrences. The scenario was initiated by a turbine trip from a full power operating conditions at three minutes into the simulation run. The following branching rules were considered:

- A hardware failure event involving a steam dump valve failure. The steam dump system is used to remove excess heat form the primary coolant system. However, if a steam dump valve sticks partially open, the continuous bleeding of steam from the secondary plant will result in an uncontrolled cooldown of the reactor plant. If the operator recognizes the steam dump failure, they may attempt a recovery action to either shut or isolate the failed valve.
- Following a reactor trip, the steam demand caused by auxiliary equipment or leakage may result in an uncontrolled cooldown of the reactor coolant system. The continued reactor cooldown may trigger a knowledge based action to shut the mains steam isolation valves (MSIVs) to prevent excessive plant cooldown. This manual closure of MSIVs due to excessive plant cooldown has been observed at nuclear power plants in the United States [98]. This action is also a possible recovery action that can be used to mitigate the plant cooldown if actions to recover the failed steam dump valve are unsuccessful. Although closure of the MSIVs will terminate the cooldown for this example, it isolates the normal condenser heat release path for the plant and may further complicate the reactor trip recovery. For example, the core decay heat must be removed by directly venting from the SGs to atmosphere.

The operations crew preference was set to enable the use of procedures (i.e., the "following procedure" problem solving strategy was activated). These analysis conditions generated the dynamic event tree shown in Figure 38.

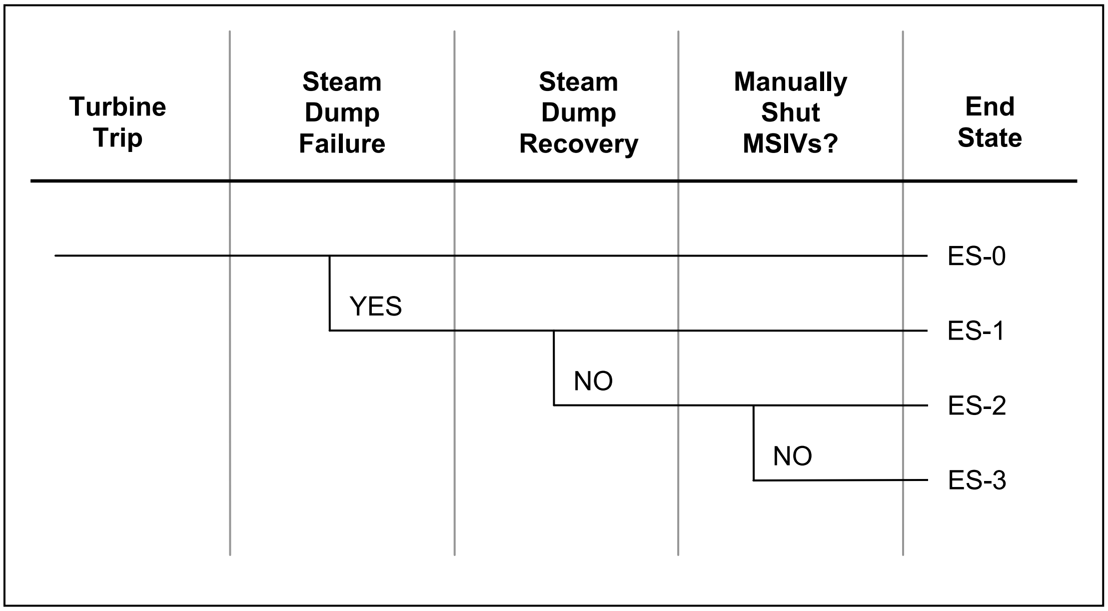


Figure 38 - Reactor Trip Dynamic Event Tree

The simulation was run for a simulated time of one hour, and four event sequences (ES) were generated. Sequence ES-0 represents the uncomplicated base case. Sequence ES-1 represents a case were a steam dump valve failed partially open following the reactor trip, but was quickly recovered by the operators. Sequences ES-2 and ES-3 are more challenging events were initial operator recovery actions for the steam dump valve fail, but manual action to shut the main steam isolation valves can isolate the steam leakage. The operators manually shut the MSIVs during sequence ES-2 but fail to do so during sequence ES-3. During sequence ES-3, the continued leakage of steam through the failed and unisolated steam dump valve eventually reduces steam pressure to the setpoint of the automatic MSIV closure safety feature. This safety feature eventually isolates the steam leakage. The timing of key events during these scenarios is summarized in Table 17.

Table 17 - Reactor Trip Scenario Event Timing Summary

Event	ES-0 (Base Case)	ES-1 (Steam Dump Recovery)	ES-2 (Manual MSIV Isolation)	ES-3 (Automatic MSIV Isolation)
Turbine Trip / Reactor Trip	181	181	181	181
Steam Dump Valve Failure	-	182	182	182
Start E-0	252	252	252	252
Start ES-0.1	328	344	344	344
Automatic RCS Letdown Isolation	-	-	419	419
Reduce AFW Flow	417	314	314	314
Recover Steam Dump	-	370	n/a	n/a
Initiate ECCS (manual due to low PZR level)	-	-	-	751
Shut MSIVs	-	-	486 (manual)	2193 (automatic)
Re-Start E-0	-	-	-	768
Start ES-1.1	-	-	-	1490
Terminate HPI Injection	-	-	-	1649
Restore Letdown	-	-	623	1786

Note: all times are provided in seconds from the start of the simulation.

Figure 39 through Figure 43 provide the response of several key plant parameters during each event sequence. As shown in Figure 39, pressurizer level immediately decreases from its nominal full power level of just below 50% to below 20% immediately after the reactor trip due to the rapid cooldown of the RCS following core shutdown. As the RCS continues to cooldown immediately following the reactor trip, the density of the reactor coolant increases and the net volume occupied by the coolant decreases causing a decrease in pressurizer level. During sequences

ES-0 and ES-1 (following recovery of the failed steam dump valve), the reactor coolant cooldown stabilizes at the hot-standby RCS temperature of approximately 547°F and the chemical and volume control system restores pressurizer level to its nominal shutdown value of roughly 22%. During ES-2, the manual closure

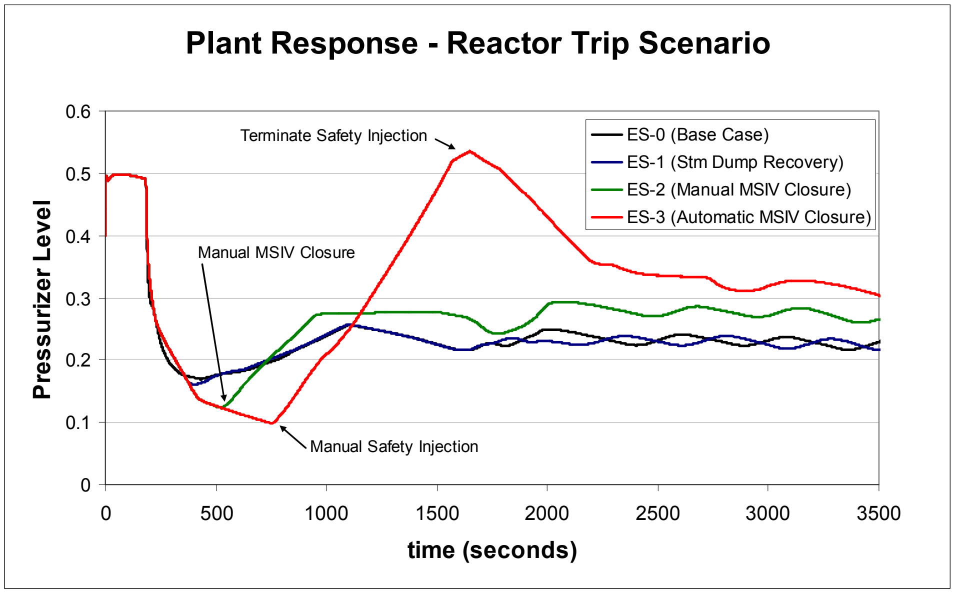


Figure 39 - Reactor Trip Scenario - Pressurizer Level Response

of the MSIVs shifts the decay heat relief path to the SG atmospheric relief valves rather than the condenser steam dump system. The higher steam pressure control setpoint of the SG atmospheric relief valves result in an approximate 20 psi higher secondary steam pressure than when using the condenser steam dump system. This results in a slightly higher average coolant temperature once conditions stabilize. Because, the pressurizer level control setpoint increases as the average RCS temperature increases, the final pressurizer level control setpoint for sequence ES-2 is slightly higher than in ES-0 or ES-1. When the operators manually actuate the safety injection system during sequence ES-3 (due to lowering pressurizer water level), the pressurizer water level begins to rapidly increase. This increase continues until operators eventually transfer to ES-1.1 which allows the crew to terminate emergency core cooling injection flow. Once injection flow is terminated, the pressurizer level continues to decrease due to continued steam flow through the failed steam dump valve. This trend continues until the MSIVs are automatically closed at 2193 seconds.

The pressurizer pressure response is shown in Figure 40. In a pressurized

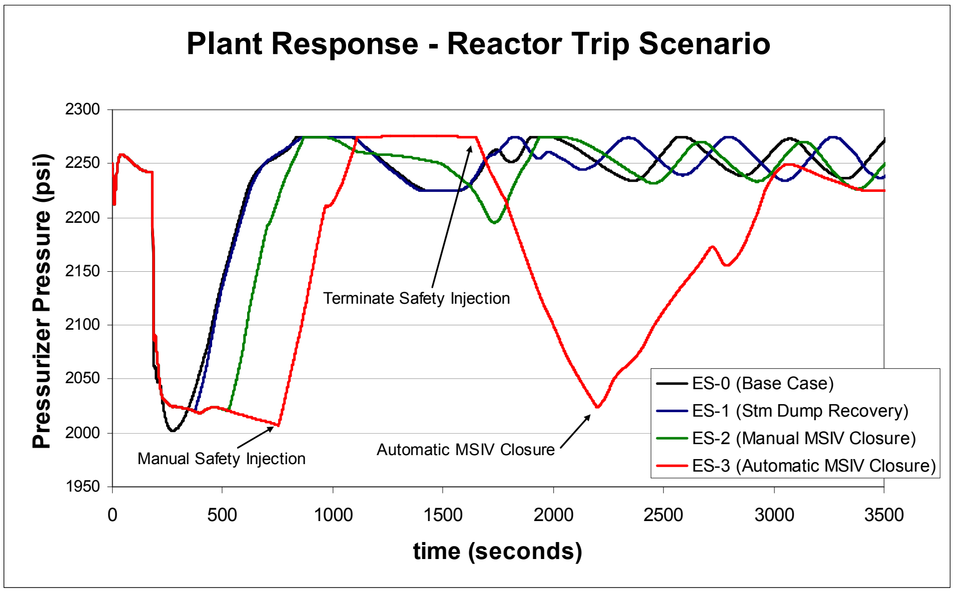


Figure 40 - Reactor Trip Scenario - Pressurizer Pressure Response

water reactor, pressure is maintained by the pressurizer. The pressurizer is a two-phase saturated water thermodynamic system and pressure is controlled through the use of a pressurizer spray system (which uses the differential pressure generated by the reactor coolant pumps) and electric heaters. The spray system cools the vapor space in the pressurizer to reduce pressure, while the electric heaters warm the liquid phase and increase pressure. The nominal pressure control setpoint for both power and shutdown operation is approximately 2250 psi. Because a decrease in pressurizer level causes the volume occupied by pressurizer vapor to increase, decreasing pressurizer water levels tend to reduce pressure. Similarly, increasing pressurize water level tends to compress the vapor bubble and increase RCS pressure. Consequently (and as expected), the pressurizer pressure tends to track pressurizer water level. This is particularly well illustrated in the case of sequence ES-3. For all scenarios, the pressurizer pressure eventually stabilizes to approximately 2250 psi. The oscillations in pressure are due to a design feature that keeps a small number of pressurizer heaters continuously energized which tends to increase pressure. The periodic cycling of the spray valve keeps pressure within the desired control band.

Figure 41 provides the RCS average loop temperature response during the four scenarios. Both the base case (ES-0) and the steam dump recovery case (ES-1), show a relatively quick stabilization to the shutdown temperature control setpoint of 547°F. Sequence ES-2 which involves manual closure of the MSIVs shows a slightly

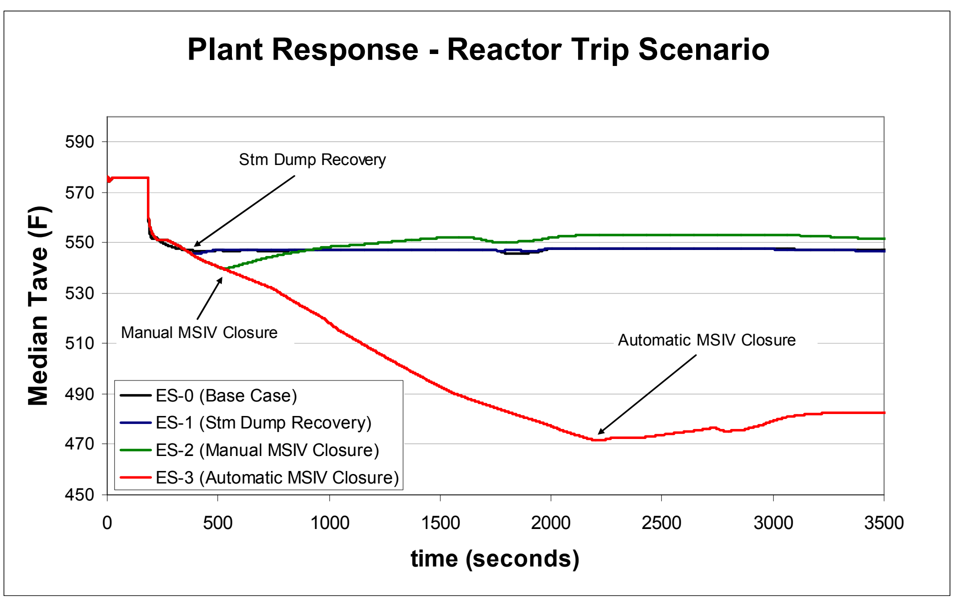


Figure 41 - Reactor Trip Scenario - Average RCS Temperature Response

higher stable temperature due to the somewhat higher pressure setpoint of the SG atmospheric relief valves. The figure allows clearly shows the continued RCS cooldown associated with the failure to recover or manually isolate the failed steam dump valve during sequence ES-4. The cooldown for this scenario is eventually terminated when the MSIVs are automatically closed at 2193 seconds.

Figure 42 provides the SG pressure response to the reactor trip scenario. During the base case (ES-0), steam pressure rapidly stabilizes to near 1000 psi. For scenarios ES-1 and ES-2, steam pressure decreases following the reactor trip until the failed steam dump valve is either recovered or isolated. Because the setpoint of the SG atmospheric relief valves is higher than the nominal steam pressure associated with the condenser steam dump system, the SG pressure stabilizes to a slightly higher value for sequence ES-2. During sequence ES-3, steam pressure continually

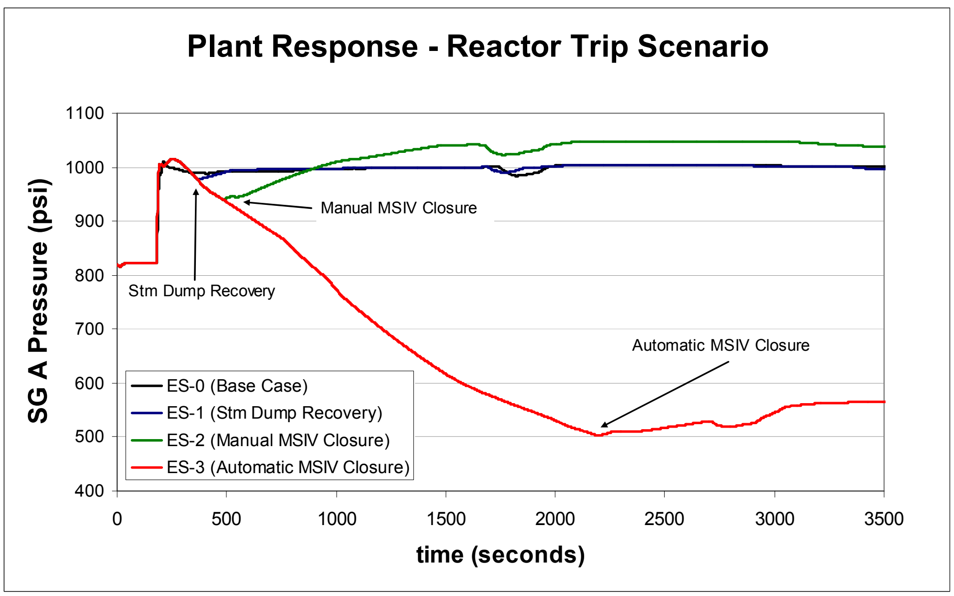


Figure 42 - Reactor Trip Scenario - SG Pressure Response

decreases at a rate of approximately 20 psi/minute until the MSIV automatic isolation safety feature setpoint is reached (approximately 500 psi). Because the SG steam pressure is tightly coupled to RCS temperature, Figure 42 show similar trends as Figure 41. Once the MSIVs are closed, the pressure reduction is terminated. During scenario ES-3, the emergency procedure E-0, step 18, directs the operators to check for a faulted SG. This check is made by verifying two conditions: (1) no SG pressure decreasing in an uncontrolled manner, and (2) no SG completely depressurized. If a faulted SG condition is observed, the operators transfer to a different emergency procedure to isolate the fault (if possible) and stabilize plant conditions. In the ADS-IDAC implementation of this step, the conditions are verified by checking that the rate of SG pressure decrease is less than 25 psi/minute and that all SG pressures are greater than 50 psi. In this case, despite clear indications that the SG pressure was decreasing in an uncontrolled manner, the step expectations could be met. This highlights that interpretation of key diagnostic steps symptom-based procedures often require a high degree of cognitive effort to correctly interpret plant conditions. Because of the wide variety of plant conditions that can exist when the operators execute this step, the application of simple decision criteria is likely insufficient to lead to a correct diagnosis. This finding is consistent with actual control room operator experience with emergency operating procedures [25].

Figure 43 provides the response of SG A water level during each scenario (the other two steam generators respond in a similar fashion). Following the reactor trip, the operators control SG water by throttle the AFW flow control valves to each SG.

The mental scheme used for water level control is similar to the model described in Section 4.3.2 (see Figure 12). These scenarios demonstrate that the mental

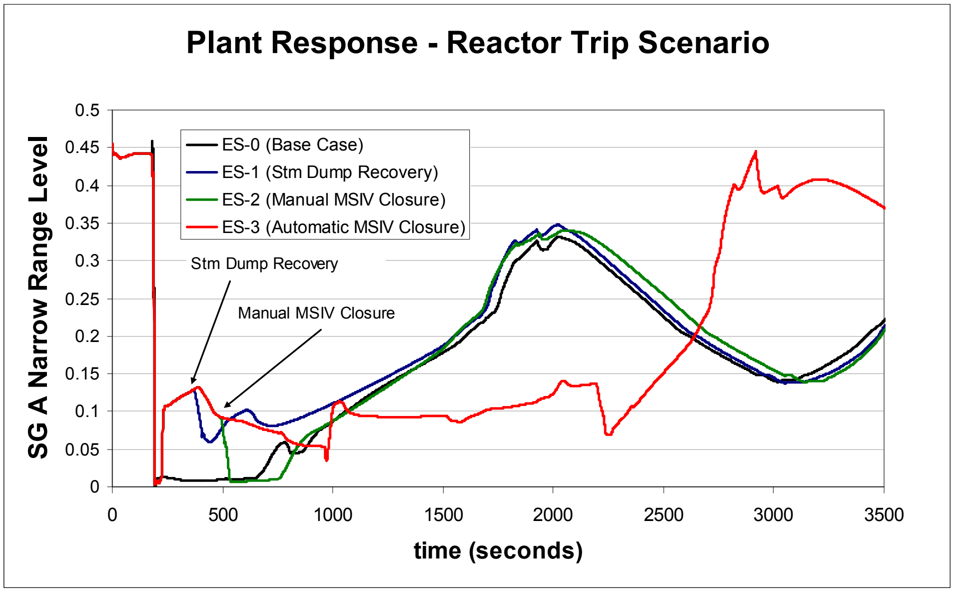


Figure 43 - Reactor Trip Scenario - SG Water Level Response

representation for AFW flow control is sufficiently robust to handle a range of potential situations. A unique feature of SG level control is that indicated water level is influenced by the steam flow from the steam generator. Steam flow tends to increase the indicated water level, while a reduction in steam flow tends to decrease the indicated level. Consequently, scenarios with a sustained steam demand from the SGs tend to have a higher indicated SG water level. The mental scheme for AFW control can account for this effect and still maintain level within the desired control band of approximately 25% level.

Despite the relative simple dynamic event tree for this scenario, the four sequences illustrate the wide range of plant responses that can be generated from the application of simple branching rules. Of particular note is that the base case scenario behaved as would be expected in an actual event of this type as demonstrated by the relatively rapid establishment of steady state conditions without exceeding normal parameter limits or causing the actuation of safety equipment. The operator recovery actions in sequences ES-1 and ES-2, generated by the activation of mental beliefs associated with the perception of excessive cooldown of the RCS, effectively terminate the adverse parameter trends and result in a stable, safe condition. This example also serves to highlight to diagnostic effort that is required to implement event symptom based procedures. As demonstrated by sequence ES-3, the diagnosis of a faulted steam generator cannot be made through the application of simple decision criteria and must depend on a holistic evaluation of plant status. Finally, it is important to note that most of the key plant parameters in a nuclear plant are tightly coupled. In this case, a failure of a single condenser steam dump valve has a dramatic impact on pressurizer level, RCS pressure and temperature, and SG water level and pressure. An advantage of the ADS-IDAC approach is that the complex relationships between physical plant parameters and operator actions are directly simulated in order to provide a rich contextual framework. A systematic modeling approach such as ADS-IDAC can help the analyst recognize, appropriately model, and gain insights about the diagnostic effort needed to effectively implement emergency plant procedures.

10.2.2 Three Miles Island Scenario

In many respects, the 1979 accident at the Three Mile Island nuclear power plant is a benchmark example of how well-intentioned actions by an operations crew can seriously degrade reactor plant safety. As discussed in Section 1.1.1, the accident was initiated by a turbine generator trip, a relatively common event. A summary timeline of the key accident events is provided in Table 18 [99]. Several complications developed as the accident unfolded, including failure of the

Table 18 - Timeline for Three Mile Island Accident

Time (minutes)	Events and Comments
0	At approximately 4 a.m. on March 28, 1979, a loss of feedwater to the steam generators resulted in a turbine trip and reactor shutdown. Shortly after the reactor trip, the pressurizer power operated relief valve (PORV) lifted (as expected) but failed to close, resulting in a loss of reactor coolant. In addition, the emergency auxiliary feedwater system which normally provides makeup water to the steam generators in order to remove residual core heat was isolated due to a system valve alignment error.
2	The two emergency core cooling high pressure injection pumps automatically began injecting core coolant in response to a low reactor coolant system pressure condition.
4.5	The operators turned off one of the two high pressure injection pumps and restricted the flow from the remaining pump in response to a high coolant level in the pressurizer.
8	The operators restored auxiliary feedwater flow to the steam generators by opening the closed isolation valves.
73	Due to the continued loss of reactor coolant from the stuck open pressurizer PORV combined with reduced high pressure injection, the reactor coolant system (which normally operates with substantial subcooling) reached saturated steam conditions. The two-phase steam and fluid mixture in the reactor coolant system led to high vibration of the reactor coolant pumps. To avoid damage to the pumps the operators turned off the B loop pumps (the pumps in the A coolant loop continued to operate).
100	The operators turned off the A loop reactor coolant system pumps due to high vibration. This terminated all forced coolant circulation through the reactor core.
111	The reactor coolant outlet temperature began to rise rapidly due to residual decay heat generation in the reactor core.
142	The operators identified that PORV valve failed to fully reclose after the reactor trip and isolated the reactor coolant system leakage by closing downstream block valve.
149	Due to the lack of forced core circulation and reduced emergency core cooling high pressure injection, the reactor temperature went off-scale at 620°F. During this period portions of the fuel cladding reached temperatures high enough to permit rapid oxidation of the zircaloy fuel cladding (an exothermic reaction that produces hydrogen gas) resulting in substantial core damage.
220	The operators restored full emergency core cooling high pressure injection flow. However, the presence of hydrogen and other non-condensable gases in the reactor coolant system hampered efforts to establish effective core cooling. Approximately 13.5 hours into the accident, the operators were able to establish subcooled reactor coolant system conditions and restart a reactor coolant pump. This permitted core decay heat to be removed via the steam generators and auxiliary feedwater system.

auxiliary feedwater(AFW) system to deliver makeup water to the steam generators (SGs) and a partially opened pressurizer power operated relief valve (PORV) which resulted in a small loss of coolant accident. Although safety systems actuated as designed, the operators misinterpreted plant conditions and terminated emergency core cooling flow prior to establishing a stable means of decay heat removal. Of particular note was that the operators strongly associated reactor coolant system water inventory with pressurizer water level. In the case of a stuck open pressurizer PORV, the pressurizer can have a high indicated level when steam bubble formation in the reactor vessel forces reactor coolant into the pressurizer. Unless the operators are aware that the reactor coolant system (RCS) has reached a bulk saturated steam condition and a steam bubble has formed in the reactor vessel, a high pressurizer level may be misinterpreted. Improved operator training and plant improvements to add more salient indications for both RCS subcooling margin and reactor vessel water level were instituted following the TMI accident to help prevent similar accidents in the future. However, the accident highlights the need for systematic methods to predict in advance situations that may lead operators to implement well-intentioned actions that lead to a serious accident.

Because one of the objectives of the ADS-IDAC project is to develop a tool capable of identifying the situational contexts that can lead to accidents such as the TMI event, a key test of model validity is the capability of the code to reproduce the conditions leading to the TMI accident. However, it should be noted that there are significant limitations with modeling this scenario in the current version of ADS-

IDAC. Although the TMI plant and the current ADS-IDAC plant model both represent pressurized water reactors, the TMI unit was a two loop Babcock & Wilcox plant design that used a significantly different reactor coolant system and SG design than the ADS-IDAC model. This project demonstrated that the key features of the event can be modeled in ADS-IDAC, however the specific accident dynamics and consequences differed.

To model the TMI event, two simulation runs were performed. The first set of scenarios included branching events for the pressurizer PORV failure and recovery and the use of either a procedure-following or knowledge-based problem solving strategy. The procedure following strategy implemented the HALDEN FRESH emergency operating procedures while the knowledge-based approach utilized the functional diagnosis approach described in Section 5.6. The specific knowledge-based actions used for this scenario are listed in Appendix F. No information filtering or biasing was introduced into the first set of scenarios. The initial simulation run generated the dynamic event tree shown in Figure 44.

Because the key operator actions associated with the TMI accident were knowledge driven and not directed by the plant procedures, the second simulation run focused on the knowledge-based problem solving strategy. In order to model the operators' lack of recognition of the degraded plant decay heat removal state due to loss of subcooling margin (i.e., the onset of saturated water conditions in the RCS), all parameter information related to subcooling margin was biased high. In other

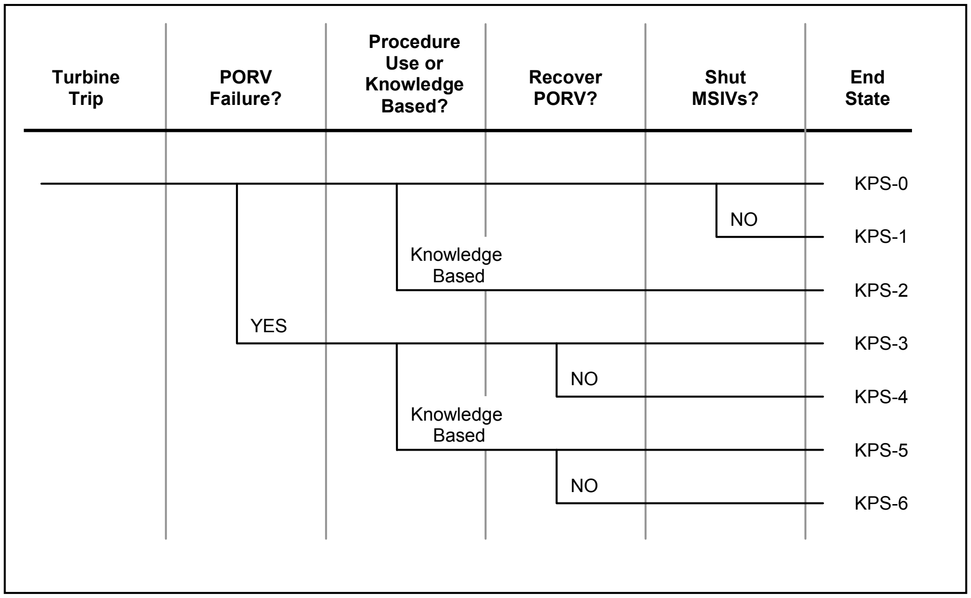


Figure 44 - Dynamic Event Tree for TMI Scenario (No Information Biasing)

words, the operators would perceive adequate subcooling margin regardless of the actual plant conditions. This approach can be used to model either an actual information perception bias, or a crew that otherwise fails to verify this indication (since a high bias will block all possible compensatory actions for this scenario). The second simulation run resulted in the dynamic event tree shown in Figure 45.

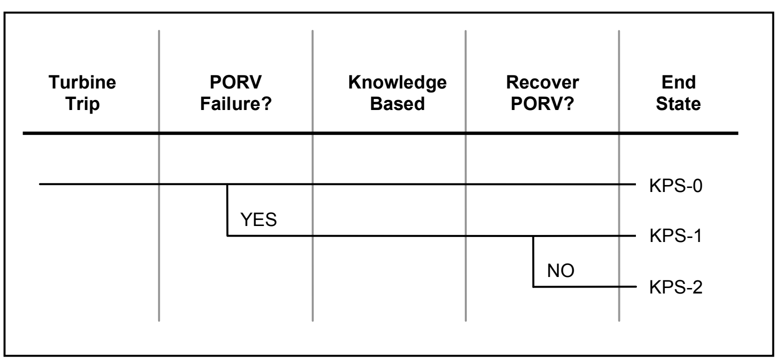


Figure 45 - Dynamic Event Tree for TMI Scenario (with Information Biasing)

Since the TMI accident involved a failure to recover from the partially opened pressurizer PORV, the following event sequences are of most interest:

- Run 1, Sequence KPS-4 – Procedure Following problem solving strategy with no information bias (Figure 44)
- Run 1, Sequence KPS-6- Knowledge Based problem solving strategy with no information bias (Figure 44)
- Run 2, Sequence KPS-2 – Knowledge based problem solving strategy with subcooling margin biased high (Figure 45)

To more realistically model the TMI event, all scenarios were initiated with AFW flow control valves in the closed position, a similar functional configuration to the TMI event. The sequence of events for these three event sequences are provided in Table 19.

Table 19 - Sequence of Events for TMI Scenario

Event	KPS-4 (Procedure Following)	KPS-6 (Knowledge Based Actions)	KPS-2 (Knowledge Based Actions w/ Bias)
Turbine Trip / Reactor Trip	181	181	181
PORV Fails (~20% open)	182	182	182
Start E-0	251	-	-
Start ES-0.1	285	-	-
ECCS Actuation	290	290	290
Return to E-0	292	-	-
Start FRG-H.1	377	-	-
Auxiliary Feedwater Flow Recovered	600	600	600
Return to E-0 (after FRG-H.1)	714	-	-
Start ES-1.1	1471	-	-
Stop 1st HPI Pump	1891	386	386
Start ES-1.2	1935	-	-
Stop 2nd HPI Pump	-	614	614
Restart 1st HPI Pump	-	1678	-
Stop all RCPs	-	1708	1746
Restart 2nd HPI Pump	-	1884	-

Note – all times are provided in seconds from the start of the simulation

During the procedure-following case, the ADS-IDAC operators immediately initiate emergency procedure E-0 and transition to supplemental procedure ES-0.1 to perform a normal recovery from the reactor trip. At the time of transition to ES-0.1, plant conditions had not degraded to the point where procedures require immediate initiation of emergency core cooling. Shortly after initiating ES-0.1, the operators perceive a low pressurizer level due to the PORV leakage (Figure 46) and manually initiate emergency core cooling. This forces the operators to return to procedure

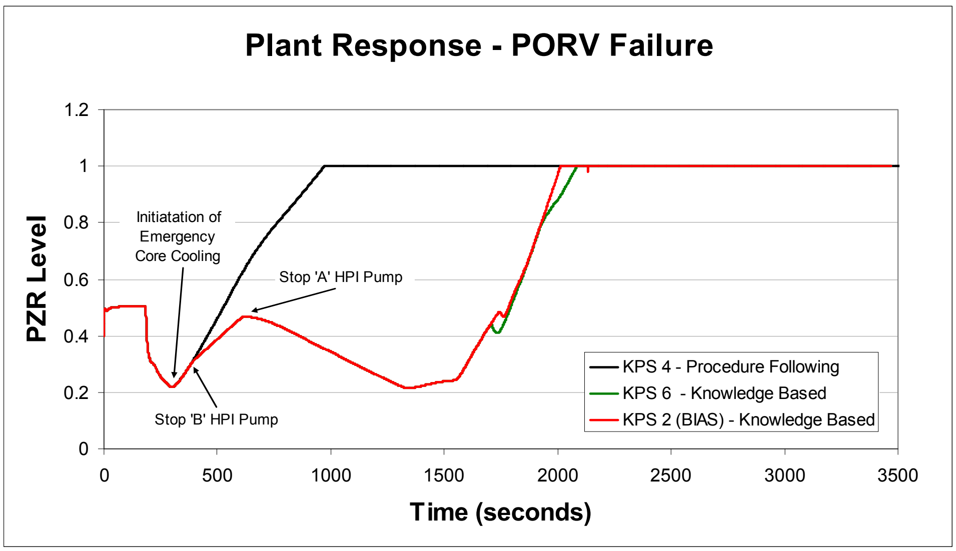


Figure 46 - TMI Scenario - Pressurizer Level

E-0. In addition, the operations crew eventually recognizes the lack of AFW flow and initiates functional recovery guideline FRG-H.1 due to a loss of secondary cooling. Initial attempts to recover AFW flow by opening the flow control valves are driven by a skill-based action that verifies the AFW valve lineup when an automatic demand for AFW is perceived. This recovery action is considered to be successful for all event sequences. Once AFW flow is restored, the operators exit FRG-H.1 and reinitiate procedure E-0. Because the net coolant leakage out of the pressurizer

PORV is less than the full capacity of the high pressure emergency core cooling system, the operators will eventually transfer to supplemental procedure ES-1.1 to terminate coolant injection in a controlled manner in order to stabilize plant conditions. The operators eventually secure one high pressure injection pump and transfer to supplemental procedure ES-1.2 to continue the plant recovery (this procedure is not modeled in the current ADS-IDAC knowledge base). A key feature of the

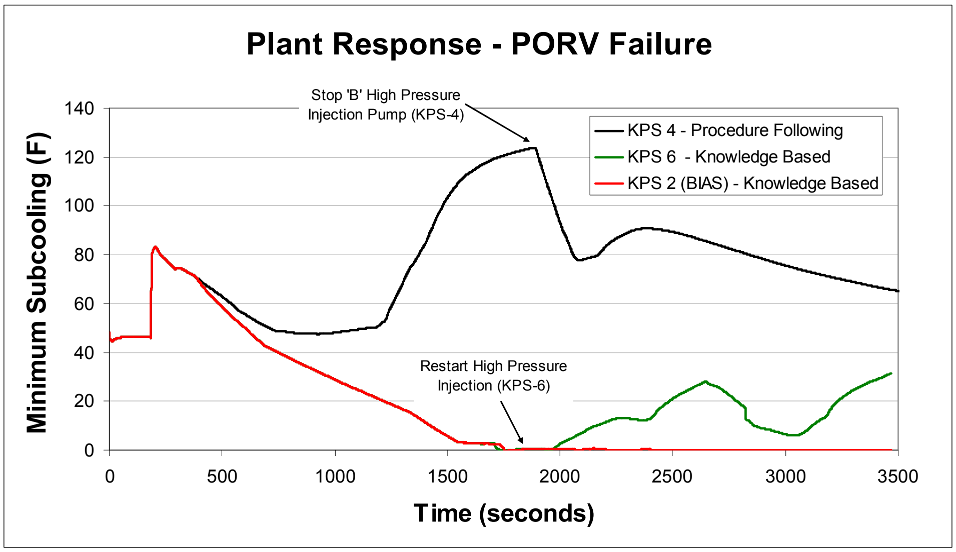


Figure 47 - TMI Scenario - Minimum RCS Subcooling

procedure based response is that an adequate subcooling margin is continuously maintained in the reactor coolant system (Figure 47). This ensures that the reactor core remains covered and core decay heat can be removed (Figure 48).

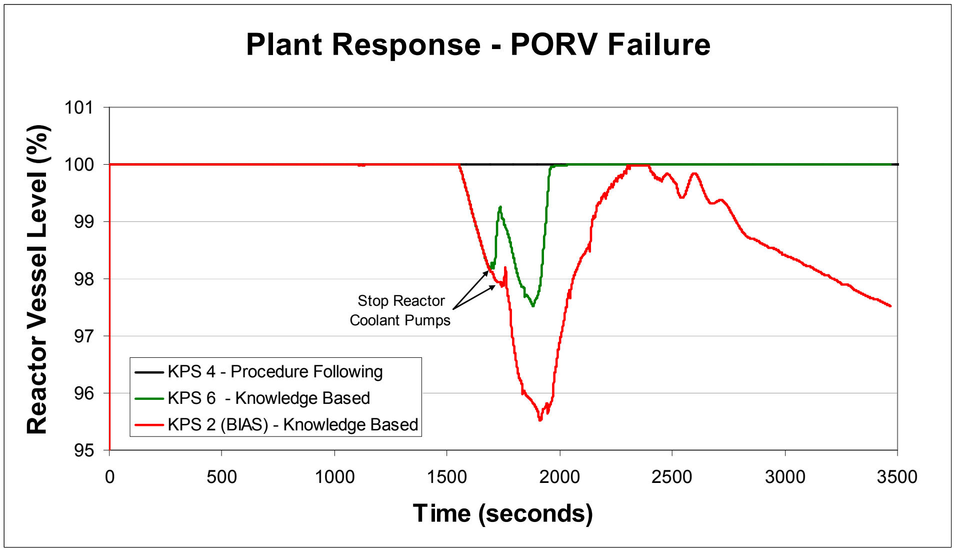


Figure 48 - TMI Scenario - Reactor Vessel Level

The knowledge-based response does not utilize the plant procedures and instead relies on a set of actions driven by the operator's situational assessment. For example, in the initial knowledge-based simulation run (KPS-6), when the operator perceives a low mass condition in the pressurizer, a knowledge based action to initiate emergency core cooling is activated. Similarly, when a high pressurizer mass condition is identified (in combination with adequate subcooling margin), a knowledge based action is activated to secure the high pressure injection pumps. Because the knowledge-based approach does not maintain a high degree of subcooling for the RCS, a two phase saturated steam/water condition is likely to develop in the reactor coolant loops. When steam voiding in the vicinity of the reactor coolant pumps exceeds a preset threshold, the thermal-hydraulic model generates a high reactor coolant pump vibration alarm. This alarm activated a skill-based operator response to stop the reactor coolant pumps to prevent cavitation damage. This behavior is similar to what was done by the TMI operators at 73 and

100 minutes into the accident (see Table 18). Conversely, this behavior does not occur during the procedure-based approach because an adequate subcooling margin is maintained and the reactor coolant pumps do not enter a degraded flow regime. Although the operators secure high pressure injection early in the accident scenario for both the knowledge-based scenarios, the continuous decrease in subcooling margin eventually prompts the operators to recognize the degraded core cooling conditions in the non-information biased case and restart high pressure coolant injection at 1678 seconds (see Figure 49). The loss of subcooling margin leads to steam bubble formation in the reactor vessel and can eventually cause uncovery of active fuel. Although the knowledge-based approach without information biasing (KPS-6) leads to a brief decrease in reactor vessel water level (Figure 48), the restoration of emergency core cooling flow restores subcooled RCS conditions and refloods the reactor vessel. Conversely, in the information biased case, the operators do not

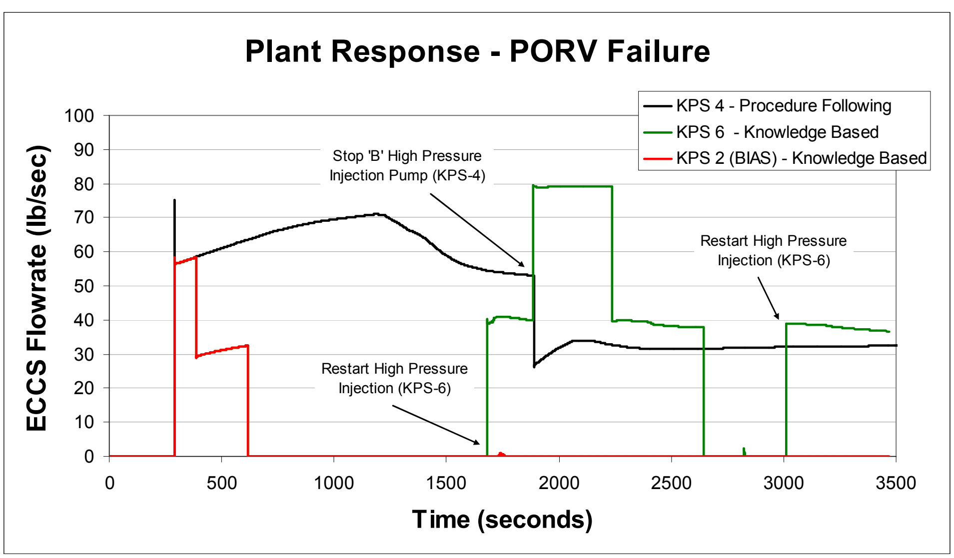


Figure 49 - TMI Scenario - ECCS Flowrate

recognize and diagnose the significance of a loss of subcooling margin and fail to restart the emergency core cooling system. This leads to a continuous decrease in reactor vessel water level as shown in Figure 48 (KPS-2). If the reactor vessel level is allowed to decrease below the level of active fuel, core damage can occur, as was the case in the actual TMI event.

This example demonstrates several key features of the ADS-IDAC modeling approach. In particular, modeling the TMI scenario highlights the following key observations:

- The ADS-IDAC model is capable of reproducing the knowledge-based behaviors observed during the TMI accident with a limited set of rules. For example, early termination of emergency core cooling flow and stopping of reactor coolant pumps can easily be modeled within the ADS-IDAC framework.
- Relatively complex scenarios can be modeled with a limited number of branching rules.
- The procedure-following model is capable of modeling complex transitions between the suite of plant operating procedures. In the procedure following case, six separate procedure transfers are executed, similar to what actual crews would need to do in this situation.
- The underlying cognitive model in ADS-IDAC allows small perturbations in information perception processes to lead to large variations in operator behaviors. For example, the only underlying difference between the two knowledge based scenarios is the biasing of subcooling margin information.

The modeling of this scenario serves to increase the level of confidence that ADS-IDAC is capable of reasonably modeling knowledge-driven errors of commission and that it can be used to identify situations that might lead to human error events.

10.3 Model Validation with HAMMLAB Data (Criterion Validity)

To further validate the capabilities of ADS-IDAC, crew behavior predictions were compared to actual crew data obtained during an international Human Reliability Analysis (HRA) empirical study at the OECD Halden Reactor Project Halden Human-Machine Laboratory (HAMMLAB) facility [100]. As discussed in Section 9.2.2, the HAMMLAB study consisted of two sets of simulator experiments, each with basic and complex variations. The first set of experiments involved steam generator tube rupture scenarios and were used to calibrate the ADS-IDAC model as described in Section 9.2.2. The second set of experiments involved loss of feedwater (LOFW) scenarios and were used to assess the predictive capabilities of the ADS-IDAC model and analysis methods. The crew compliment during the Halden experiments consisted of three crew members: a shift supervisor (a senior reactor operator), a reactor operator, and an assisting operator. The balance of plant operator position, a normal member of operating crew, was not used during the Halden experiments. Within the ADS-IDAC model, the shift supervisor is analogous to the decision-maker and the reactor operator and assisting operator are modeled by a single action-taker. Although the HAMMLAB simulator experiments were completed in late 2006, the ADS-IDAC crew performance predictions were made without knowledge of the actual crew performance during the LOFW simulator experiments (i.e., the predictions were done in a blind manner).

10.3.1 Loss of Feedwater Scenario Descriptions

During normal reactor plant operation, the steam generators use heat from the reactor coolant system to vaporize main feedwater in order to provide steam for main turbine operation. At steady-state conditions, a control system adjusts main feedwater flow to maintain the secondary steam generator water level within a specified range. If main feedwater flow is lost due either to a control system malfunction or a loss of the main feedwater pump flow, steam generator water level rapidly decreases. To prevent a loss of core cooling, the reactor and main turbine are automatically shut down when a low steam generator water condition occurs, and an alternate auxiliary feedwater system is actuated to provide cooling water to the steam generators. If the auxiliary feedwater system is unavailable following a loss of main feedwater, the operators should normally attempt to recover a source of feedwater flow or initiate an alternate means of core decay heat removal.

HAMMLAB procedure functional recovery guideline FR-H.1, "Response to Loss of Secondary Heat Sink," specifies three key operator actions following a complete loss of steam generator feedwater:

- Stop the reactor coolant pumps to minimize heatup of the reactor coolant system and extend the time available for recovery of feedwater flow.

- Attempt to recover a source of feedwater flow from either the auxiliary feedwater system, main feedwater system, or the condensate system. If the condensate system is used, secondary pressure must be reduced due to the lower pressure capability of the condensate pumps.
- Initiate reactor coolant system feed and bleed cooling if no source of feedwater can be recovered. Feed and bleed cooling is initiated by aligning emergency core cooling and opening a relief path in the reactor coolant system.

A flowchart of the key FR-H.1 operators actions included in the ADS-IDAC model is provided in Figure 50. In order to model timing variations for control room crews, three briefing holds were identified in the procedure model: (1) prior to the start of the procedure, (2) prior to depressurization of the reactor coolant system if a low pressure source of feed is available, and (3) prior to initiation of feed and bleed cooling. Although FR-H.1 does not require the operators to initiate feed and bleed cooling until the wide range level in at least two SGs is less than 12%, it was believed that some crews would initiate feed and bleed cooling earlier. Rather than basing early initiation of feed and bleed on a specific SG level, it was decided to model an early transition through the use of a "watchdog" timer. The timer monitors the elapsed time since initiation of FR-H.1 and transitions to feed and bleed when the timer exceeds a preset threshold value. Thus, either low SG wide range level or a fully elapsed watchdog timer can initiate a transition to feed and bleed cooling. It was also noted that operator actions during FR-H.1 step 7 to permit SG feeding from a low pressure water source may lead to an inadvertent actuation of the emergency

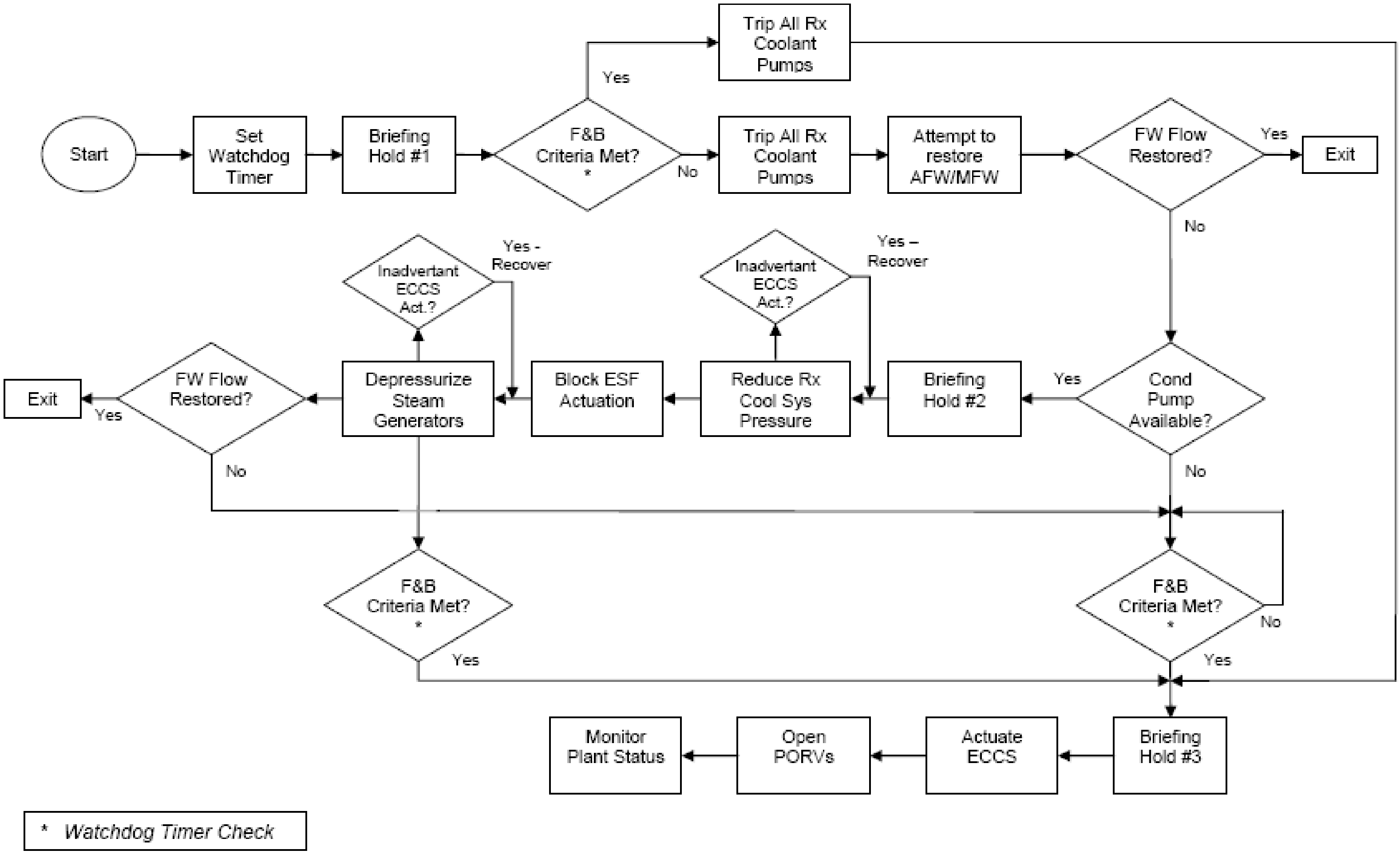


Figure 50 - Initial ADS-IDAC FR-H.1 Procedure Model

core cooling system. Specifically, step 7.a directs the operators to reduce RCS pressure to just below the permissive setpoint for the low pressurizer pressure and high steam flow safety injection feature. An excessive amount of depressurization prior to blocking these signals (or allowing RCS pressure to increase above the block reset setpoint) could result in an inadvertent safety injection. Similarly, actions during step 7.c to depressurize one or more SGs to a pressure below the condensate pump discharge pressure could result in the generation of a safety injection if the cooldown is not well controlled.

10.3.1.1 LOFW Scenario Base Case

The base case loss of feedwater scenario begins with a complete loss of condensate pump flow with the reactor plant operating at 100% power. The loss of condensate flow causes the main feedwater pumps to trip on low suction pressure resulting in a complete loss of feedwater to the steam generators. Steam generator levels then rapidly decrease and the reactor automatically trips approximately 20 seconds after the start of the scenario due to low steam generator water level. If the operators quickly identify the loss of feedwater condition, they may initiate a manual reactor trip prior to automatic trip and conserve steam generator secondary inventory. Following the reactor trip, both motor driven auxiliary feedwater pumps (MDAFPs) and the turbine driven auxiliary feedwater pumps (TDAFP) are assumed to fail. The operators are expected to diagnose that a complete loss of main and auxiliary feedwater has occurred and initiate procedure FR-H.1, “Response to loss of secondary heat sink." Because flow from the condensate pumps, the MDAFPs, and the TDAFP cannot be recovered, the control room operators will need to initiate primary feed and bleed decay heat removal when the wide range level in two steam generators decreases below 12%. If feed and bleed cooling is not promptly established, the reactor core heat generation will increase the reactor coolant system pressure and eventually cause the pressurizer PORVs to open. In order for feed and bleed cooling to provide adequate heat removal, it must be established before the mass flow rate through the pressurizer PORVs exceeds the injection capability of the high head safety injection pumps. Furthermore, core damage may occur if the injection flow is insufficient to prevent the reactor vessel water level decreasing below the level of the nuclear fuel.

10.3.1.2 LOFW Scenario Complex Case

The complex loss of feedwater scenario begins with a partial loss of condensate system flow with the reactor plant operating at 100% power. The partial loss of condensate is initiated by the failure of all but one condensate pump. However, the remaining condensate pump is degraded with a maximum discharge pressure of approximately 360 psi (normal discharge pressure for the pump is approximately 500 psi). The low condensate discharge pressure causes the main feedwater pumps to trip on low suction pressure and results in a complete loss of feedwater. Steam generator levels then rapidly decrease and the reactor automatically trips approximately 20 seconds after the start of the scenario due to low steam generator water level. If the operators quickly identify the loss of feedwater condition, they may initiate a manual reactor trip prior to automatic trip and conserve steam generator secondary inventory. Following the reactor trip, both motor driven auxiliary feedwater pumps (MDAFPs) and the turbine driven auxiliary feedwater pumps (TDAFP) are assumed to fail.

The operators are expected to diagnose that a complete loss of secondary makeup has occurred and initiate procedure FR-H.1, "Response to loss of secondary heat sink." With a condensate pump is available (but degraded), the operators may attempt to align feedwater flow to the steam generators from the condensate system. Because steam generator pressure is significantly greater than the discharge pressure of the condensate pump, the operators will need to increase steam flow in order to reduce steam generator pressure. If the operators successfully align the condensate system to the steam generators before reaching the initiation criteria for feed and bleed cooling, FR-H.1 can be terminated once the narrow range steam generator levels are above 10%. Although steam generator depressurization can lead to successful recovery of feedwater flow, higher steam flow reduces the steam generator water inventory faster and reduces the time until feed and bleed cooling must be initiated. An additional complication is that two steam generator wide range level instrument failures mask the transfer criteria to feed and bleed. Specifically, the wide range level indicator for the A steam generator becomes stuck at a value of 16% and the steam generator C indicator reads 15% high (i.e., a constant +15% bias). Because the FR-H.1 Step 9 transition criteria for feed and bleed cooling requires that two wide range steam generator level indicators are below 12%, these instrument failures might delay or prevent the crew from initiating the appropriate transition.

10.3.2 ADS-IDAC LOFW Predictions

Based on a detailed review of the HAMMLAB loss of feedwater scenario, the associated Halden emergency procedures, and relevant industry operating experience, four main categories of crew variability were identified:

(1) diagnosis and situational assessment;

(2) timing;

(3) procedural adherence; and

(4) control inputs and manipulations.

These categories are associated with the following crew behaviors:

- Diagnosis and Situational Assessment: Immediately following the loss of feedwater event, an alert crew may quickly diagnosis the cause of the event and initiate a manual reactor trip before an automatic trip is actuated. An earlier reactor trip preserves water inventory in the steam generators and extends the time available until feed and bleed cooling must be initiated. Diagnosis of a loss of heat sink condition requires the crew to perceive low steam generator water levels coincident with low feedwater flow. Although the emergency procedures eventually prompt the operators to verify these parameters, a situationally aware crew might be expected to monitor these parameters and execute an earlier transition to FR-H.1, “Response to loss of secondary heat sink.”

- Timing: The ADS-IDAC model included four main sources of timing variability. These include:

  (1) the transition time between recognizing loss of secondary heat sink and initiating FR-H.1;

  (2) the time between initiation of FR-H.1 and tripping the reactor coolant pumps (FR-H.1 Briefing Hold #1);

  (3) the time between initiation of FR-H.1 and start of secondary depressurization (FR-H.1 Briefing Hold #2); and

  (4) the time between satisfying criteria for establishing of feed and bleed cooling and actual initiation of feed and bleed (FR-H.1 Briefing hold #3).

  Crews may exhibit timing variations due to the conduct of crew briefings or differences in procedure execution approaches that result in faster or slower procedure pacing.

- Procedure Adherence: ADS-IDAC includes a procedure step-skipping model. The probability of skipping a step action is dynamically calculated based upon the baseline skip probability for the step, the type of procedure being followed, the step objectives, the relevance of the action to the operator's situational assessment, and certain performance influencing factors. The results from the ADS-IDAC step-skipping model for a representative complex case scenario are provided in **Error! Reference source not found.**. As can be seen in the figure, tripping the reactor coolant pumps as required by FR-H.1, Step 3, and failing to block automatic safety injection actuation in FR-H.1, Step 7.4 were highlighted as actions that may be omitted by the crew. The failure to trip the reactor coolant pumps increases the heat input into the reactor plant and reduces the time available until feed and bleed cooling must be initiated. The failure to block safety injection may result in an inadvertent emergency core cooling system actuation which isolates the main feedwater system and complicates restoration of a feedwater source to the steam generators.

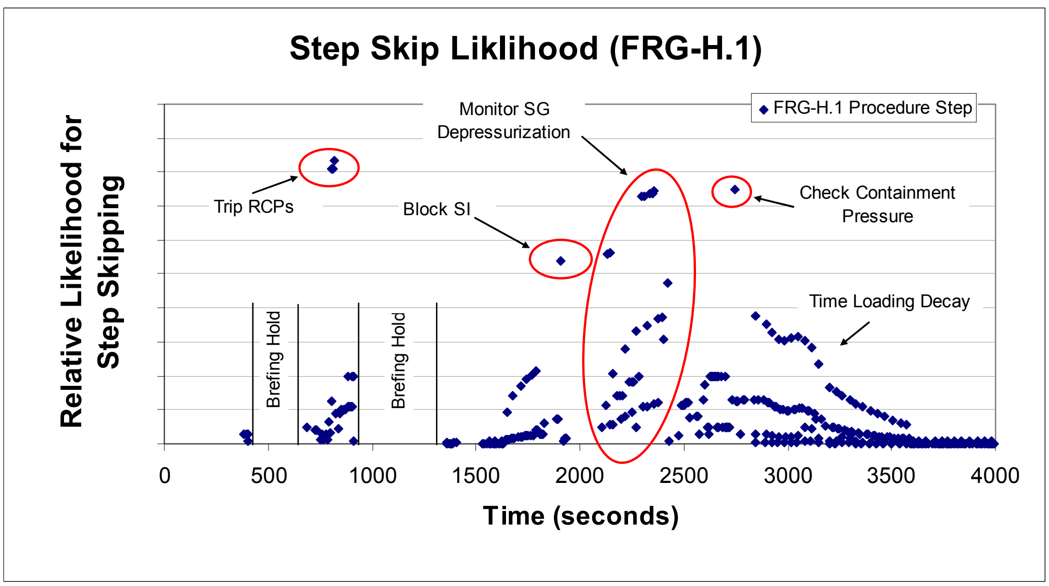


Figure 51- Predicted Step-Skipping (Loss of Feedwater Scenario)

During SG depressurization in the complex case, the need to continuously monitor SG pressure combined with time pressure may result in some crews failing to adequately control the depressurization. This may lead either to an excessive rate of depressurization (which may inadvertently actuate safety features) or an excessive amount of depressurization (which may delay restoration of feedwater).

- Control Manipulations: Certain procedure steps require operators to manipulate an adjustable control to achieve a specified control setting. Typical examples include changing setpoints for automatic control systems or adjusting regulating valves. Control manipulations where operators may adjust control settings too little or too much (e.g., opening a regulating valve either more or less than desired) are potential sources of crew-to-crew variability. Three main sources of control variability were identified in the loss of feedwater emergency procedures: (1) improper selection of reactor coolant system target pressure during FR-H.1, Step 7.a, leading to an inadvertent safety injection; (2) use of an excessive cooldown rate during the FR-H.1, Step 7.c, leading to an automatic main steam isolation; and (3) failure to open all available pressurizer power operated valves (PORVs) during FR-H.1, step 15.

Specific Predictions

Using the guidance outlined in Section 9.3, a review of Halden emergency procedures, and applicable industry-wide operating experience, the following potential sources of crew-to-crew variability were identified:

- Early or Late Reactor Trip: If the control room immediately recognizes that a complete loss of feedwater has occurred, they may take action to manually trip the reactor. If the operators fail to immediately diagnosis the LOFW event, the reactor will be automatically tripped due to a low-low steam generator water level condition. Initiating a manual reactor trip will preserve inventory in the steam generators and provide additional time until feed and bleed cooling must be initiated.

- Detection of Low Auxiliary Feedwater Flow: The operators are expected to initiate FR H.1 when a low auxiliary feedwater condition occurs coincident with a low level in all steam generators. The recognition of low auxiliary feedwater flow can either be self-identified by the crew or prompted by the emergency procedures. For both the base and complex scenarios, it is expected that the crew transfers to ES-0.1 after verifying that safety injection is not required in E-0, Step Once the crew transfers to ES-0.1, they should commence monitoring critical safety functions (Halden procedure F-0, "Critical safety function status trees") in accordance with E-0 foldout page instruction 3 and ES 0.1 foldout page instruction 3. The operators are prompted to verify auxiliary feedwater flow by the following steps:
    1. ES-0.1, Step 6 requires the operators to verify steam generator water levels and feedwater flow;
    2. Status Tree F-0.3, "Heat Sink," requires the operators to assess both steam generator water levels and total feedwater flow.

  If the operators fail to transfer to ES-0.1 following E-0, Step 4, there are several additional prompts to verify feedwater flow:

1. E-0, Step 6, requires that the operators verify the status of the auxiliary feedwater pumps. Because all auxiliary feedwater pumps have failed during the scenario, the operators would be expected to recognize the low auxiliary feedwater flow condition.
2. E-0, Step 12, requires the operators to verify total auxiliary feedwater flow and specifically directs the operators to transition to FR H.1 if a low flow condition is detected.
3. E-0, Step 22, directs the operators to begin monitoring critical safety function status trees. This may prompt the operators to verify feedwater flow in accordance with the heat sink status tree (F-0.3).

In addition to being directly prompted by a procedure, the crew may self-identify the low auxiliary flow condition based on control panel monitoring and diagnostic activities. Self-identification of the loss of auxiliary feedwater may result in an earlier initiation of FR H.1. Consequently, the operators may trip the reactor coolant pumps sooner and extend the available time until feed and bleed cooling must be initiated.

- Failure to Stop Reactor Coolant Pumps: FR-H.1, Step 3, directs the crew to trip all operating reactor coolant pumps to reduce the reactor coolant system heat input and extend the time available until feed and bleed must be initiated. Because the operators may not immediately recognize the benefits of tripping the reactor coolant pumps, it is possible that the crew may skip this step. This will result in a decreased available time until feed and bleed cooling must be initiated.

- Inadvertent Safety Injection During Reactor Coolant and Steam Generator Depressurization: During the complex scenario, FR-H.1, Step 7, requires the crew to reduce reactor system pressure below 2015 psi in order to block the low pressure safety injection signal. However, if the operator reduces reactor coolant system pressure below the safety injection setpoint (~ 1845 psi for the ADS-IDAC reference plant) prior to blocking the SI signal, the safety injection system may automatically actuate. Similarly, if the crew fails to block automatic safety injection actuation or allows the block to reset due to reactor coolant system pressure increasing above 2015 psi, the safety injection system may actuate if later steam generator depressurization causes reactor pressure to decrease below the safety injection setpoint. A safety injection actuation will increase reactor coolant system inventory and pressure and cause isolation of the main feedwater system. Therefore, the operators will need to interrupt ongoing activities to terminate safety injection and reset the main feed water isolation.

- Threshold Criteria for Initiation of Feed and Bleed: FR-H.1, Step 9, directs the crew to initiate feed and bleed cooling when the wide range level in two steam generators is less than 12% or reactor coolant pressure exceeds approximately 2390 psi. However, based on their assessment of the plant status, the crew may elect to initiate feed and bleed early, particularly if they determine that a source of feed water flow cannot be immediately restored. During the complex case, the transition to feed and bleed cooling may be delayed because two of the wide range steam generator level indicators are stuck at a level above 12%. Based on the crew's experience and training, the time required to recognize the failed wide range level instrumentation and transition to feed and bleed cooling may vary.

In order to facilitate the modeling of variations in the feed and bleed transition criteria, a watchdog timer was introduced to trigger initiation of feed and bleed following a specific time lag from the start of FR-H.1. For the base case scenarios, the watchdog timer is used to force an early transition to feed and bleed (i.e., the operators initiate feed and bleed when the watchdog timer expires, regardless of SG wide range level). Because all SG wide level instruments are functional in the base case, feed and bleed will also be initiated when two SG wide range indicators reach 12% (regardless of the watchdog timer setting). During the complex case scenarios, two of the three wide range SG level instruments are stuck at a value above the feed and bleed transition criteria. Consequently, the procedural requirement for feed and bleed initiation based on SG level will never be satisfied. The watchdog timer is used to trigger the transition to feed and bleed (i.e., the operators initiate feed and bleed when the watchdog timer expires, regardless of the actual or perceived SG wide range level).

- Reactor Coolant System Bleed Path: FR-H.1, Step 15, requires the operators to open all three pressurizer power operator valves (PORVs). However, because opening all PORVs will result in a rapid depressurization of the reactor coolant system, the crew may elect to open the PORVs one at a time or open fewer than three PORVs. However, the failure to establish an adequate bleed path may limit the effectiveness of feed and bleed cooling and result in a greater likelihood of core uncovery.

- Procedure Pacing and Timing: In order to model variations in the pace and timing of procedure execution, three procedure hold points were identified:
    1. Prior to initiation of FR-H.1, Step 1;
    2. Prior to reactor coolant system depressurization in FR-H.1, Step 7; and
    3. Prior to feed and bleed cooling initiation in FR-H.1, Step 10.

  In the absence of any other source of available timing data, the length of all procedure hold points was modeled by a three parameter Weibull probability density function with a scale parameter of 200.0 seconds, a shape factor of 1.75, and a minimum time of 0.0 seconds. These parameters were set based on experience obtained during Phase 1 of the empirical study for the steam generator tube rupture scenarios. The length of these three briefing holds represents a significant source of uncertainty for the LOFW predictions.

A detailed summary of the sources of crew-to-crew variability is provided in Appendix G, "LOFW Scenario: Predicted Branching Events." LOFW predictions were documented in a report [101] that was provided to Halden and U.S. NRC staff associated with administration of the HRA empirical study.

10.3.4 Preliminary Comparison to HAMMLAB Experimental Results

After a set of operator behavior predictions for both the base and complex scenarios were documented and provided to the International HRA Empirical Study administrative team, the actual simulator log data was reviewed to determine the accuracy of the ADS-IDAC predictions. Although the simulator log data provides a wealth of information about the behavior of nuclear plant systems and operator control inputs, it lacks qualitative data that could be used to better calibrate and validate the model. For example, the simulator log data contains a detailed time history of plant thermal-hydraulic parameters, alarms, and control inputs. However, it is difficult to determine when operators commenced a procedure or held a crew briefing or meeting, and one can never be certain of crew motivations for behaviors. This information must be inferred from the more objective and quantitative log file data. Despite this limitation, the blind prediction process used for this study provided a number of valuable insights about the modeling requirements needed to accurately predict operator behaviors.

10.3.4.1 General Analysis Approach for Experiment Results

Similar to the data analysis approach utilized during the calibration process (see Section 9.2.1), the simulator log data for the LOFW scenarios was reviewed to identify key benchmark points that could be traced back to explicit procedure steps. These benchmarks were then used to determine crew timing variability and areas where crews deviated from procedural requirements. Based on a comparison between the governing Halden emergency operating procedures and the log data files, the following benchmark points were identified:

- Alarm Actuation: Low Main Feed Pump Suction Pressure. The main feed pumps automatically trip when a low suction pressure condition occurs. Therefore, this alarm is defines the initiating loss of feedwater condition for both the base and complex scenarios.

- Alarm Actuation: Reactor Trip Breaker Open. This alarm is actuated when either the crew initiates a manual reactor trip or an automatic reactor trip is activated due to low SG water level. By reviewing the time difference between actuation of the main feed pump suction and the reactor trip alarm, the associated SG water levels, and operator control inputs; it is possible to determine if the reactor trip was manually or automatically generated.

- Alarm Actuation: Low Flow – Reactor Coolant Pumps 1-3. This alarm is activated when the operators stop each loop reactor coolant pump. Step 3 of procedure FR-H.1 directs the crew to stop all reactor coolant pumps. Therefore, this alarm provides an indication of when each crew performs this procedure step.

- Component Status: Opening of RCV227VP (Pressurizer Auxiliary Spray) or RCP052VP, RCP051VP, RCP050VP (Pressurizer PORVs). In order to block the actuation of the low pressurizer pressure safety injection safety feature, it is necessary for the operators to depressurize the RCS to a pressure below the blocking permissive setpoint. Step 7.a. of FR-H-1 specifies two methods to accomplish this depressurization – use of the auxiliary spray system or opening of a pressurizer PORV (the normal spray system cannot be used since the reactor coolant pumps are stopped in step 3 of FR-H.1). This action is only performed if the crew believes it will be possible to align a low pressure feed source (such as the condensate pumps) to provide SG makeup flow.

- Alarm Actuation: Low Reactor Coolant Pressure Safety Injection Blocked. This alarm is activated when the operators intentionally block actuation of the low pressurizer pressure emergency core cooling safety feature. The operators can only block this feature if reactor pressure has first been reduced below the permissive setpoint. Procedure FR-H.1, step 7.b directs the crew to perform this action.

- Alarm Actuation: High Steam Flow Safety Injection Blocked. This alarm is activated when the operators intentionally block actuation of the high steam flow emergency core cooling safety feature. The operators can only block this feature if certain permissive conditions have been met. Procedure FR-H.1, step 7.b directs the crew to perform this action.

- Parameter Value: Wide Range SG Level in 2 SGs < 12%. As specified in caution preceding step 2 of FR-H.1 and explicitly required by FR-H.1, step 9, the operators are directed to initiate feed and bleed cooling when the wide range level in at least two SGs decreases below 12%. For the base case, all SG wide range level instruments are accurate and this condition can be directly determined. For the complex case, the SG A and C wide range level indicators were biased high and this condition is inferred based on the level in SG B (there is a presumption that the level in all SGs decreases at approximately the same rate).

- Alarm Actuation: Manual Safety Injection Actuated. In order to establish feed and bleed decay heat removal, step 10 of FR-H.1 directs the operators to actuate a manual safety injection. This action will activate the manual safety injection alarm. The time delay reaching the entry condition for feed and bleed cooling (two SGs less than 12% wide range level) and activation of this alarm provides an indication of the operators situational assessment and whether the crew delays in performing this required action.

- Component Status: Pressurizer PORV Valves Opened (RCP052VP, RCP051VP, RCP050VP). Step 15 of FR-H.1 directs the operators to open all available pressurizer PORVs. This step completes alignment of the feed and bleed decay heat removal path.

These benchmark points, and any intervening operator actions, can be mapped for each crew to determine specific timing and areas of human performance variability. In addition to determining the timing of these key benchmarks, the behavior of key plant parameters (e.g., SG water levels and pressure, RCS temperature and pressure, and pressurizer water level) was plotted for each crew to determine both the plant context for operator actions and if any unanticipated actions were performed.

10.3.4.2 LOFW Base Case Scenario Comparison to Predictions

For the relatively straightforward base scenario, it was determined that several predicted behaviors were observed during the Halden experiments. For example, manual action to perform an early reactor trip, certain timing variabilities, and early transition to feed and bleed cooling were both predicted and observed for the base scenario. However, it was determined that the ADS-IDAC predictions did not adequately capture all the plant behavior and timing variability among the Halden experiment crews. For example, Figure 52 provides a comparison between Halden

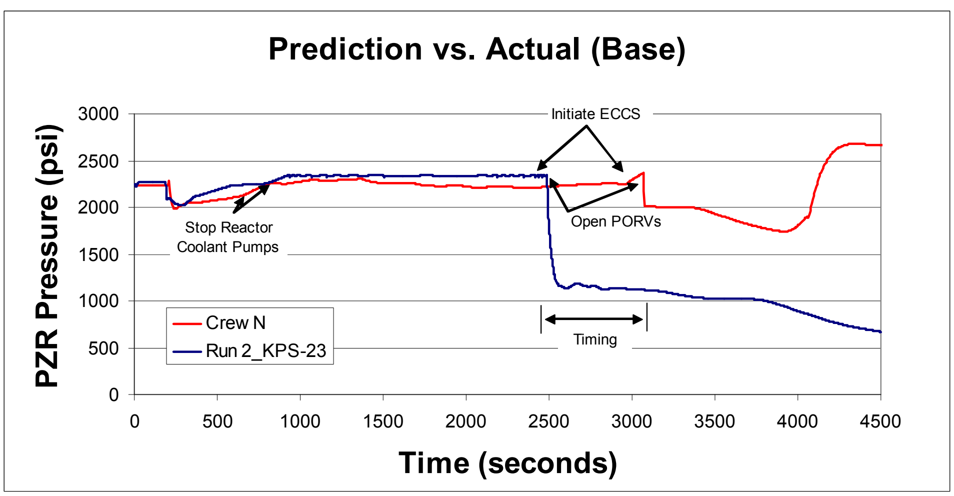


Figure 52 - PZR Pressure Response (Base - Crew N)

Crew N and a comparable ADS-IDAC predictive simulation run. Although certain aspects of the scenario are similar between the two cases (e.g., time to trip reactor coolant pumps and overall plant behavior) the timing for initiation of feed and bleed is significantly different. For this crew, the transition to feed and bleed cooling was based on reaching the procedural requirement of at least two SGs with a wide range level less than 12%. The same transition criterion was used for the ADS-IDAC prediction. However, as shown in Figure 53, there is a significant calibration difference between the wide range level indicator for the FRESH simulator and

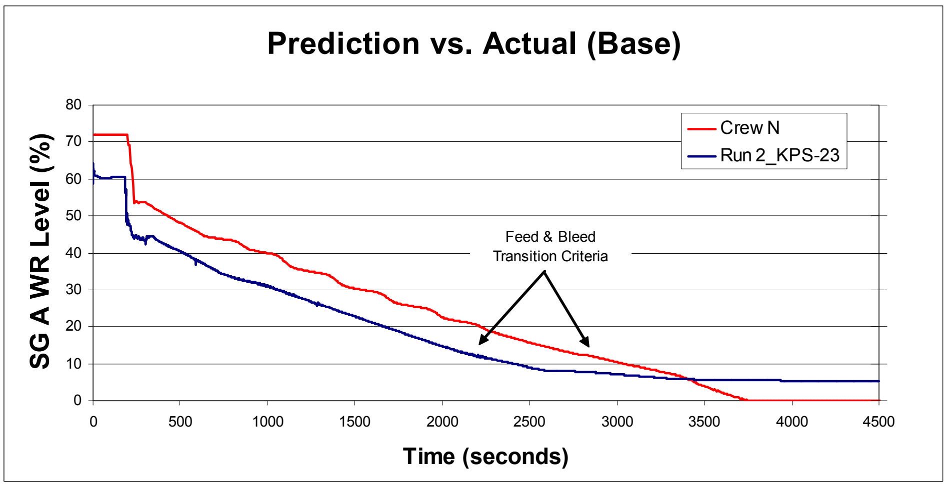


Figure 53 - SG Water Level Response (Base - Crew N)

in the ADS-IDAC model. Consequently, even though both sequences follow similar behavior rules, the actual transition time to feed and bleed cooling is dramatically different due to the ADS-IDAC model reaching a 12% wide range level condition approximately ten minutes before the FRESH simulator. Additionally, the pressurizer pressure decrease for the Halden crew following initiation of feed and bleed appears to stop at approximately 2000 psi. Upon consultation with Halden Reactor Project staff, it was determined the pressure behavior was due to an additional equipment failure that was included in the experimental scenario but not initially provided to the empirical study participants. Specifically, when pressurizer pressure decreased below approximately 2000 psi, the pressurizer PORV failed closed resulting in loss of the bleed flow path.

The combined impact of the time to perform an early reactor trip and the time required to stop the reactor coolant pumps was also investigated. Tripping the reactor early in the accident reduces water loss from the SGs and maximizes the availability water inventory for decay heat removal. Tripping the reactor coolant pumps eliminates a significant heat input into the reactor plant and increases the time that heat removal via the SGs can be used to maintain core cooling. An evaluation of the Halden data indicated that strong relationship between reactor trip time and available time until feed and bleed cooling had to be initiated (Figure 54). Crews that initiated feed and bleed prior to reaching the 12% wide range SG level criterion were excluded from this analysis. Based on this data analysis, tripping the reactor eight seconds

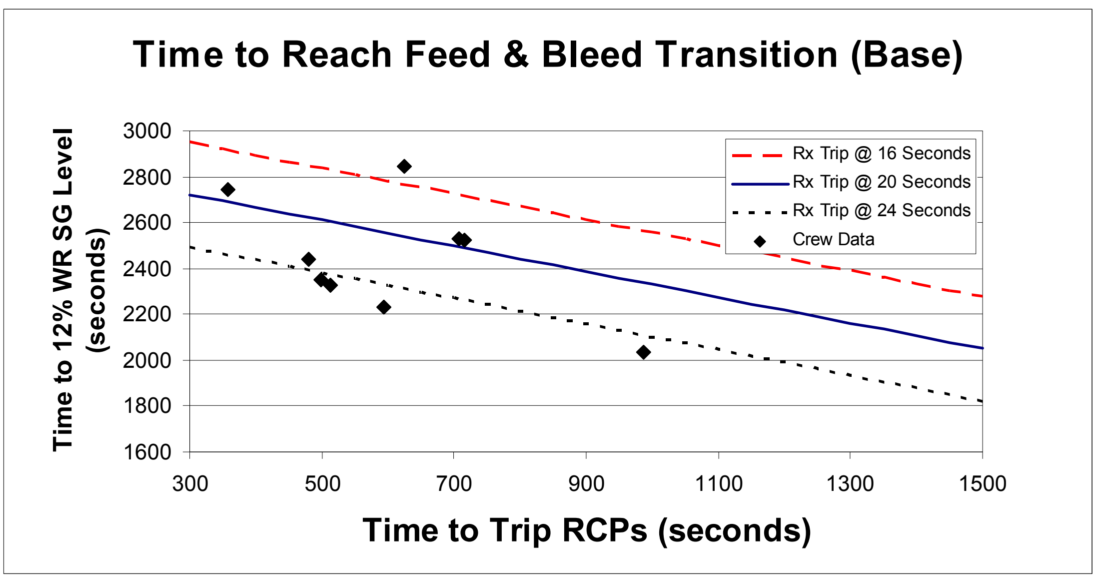


Figure 54 - Influence of Reactor and RCP Trip Time on Feed & Bleed

earlier increases the time until feed and bled must be initiated by approximately eight minutes. Although the influence of reactor coolant pump trip time is less strong, tripping the pumps ten minutes later reduces the available time until feed and bleed by roughly five minutes. These general trends are consistent with those calculated by the ADS-IDAC model. This gives rise to the interesting conclusion that a crew who can quickly diagnosis the accident and expeditiously perform the initial procedure steps can substantially increase the time available until feed and bleed cooling must be initiated. This serves to underscore one of the chief benefits of a dynamic simulation model in that the influence of prior operator actions and contextual factors are explicitly considered within the model.

Because of the nature of the base scenario, only procedure hold #1 (i.e., prior to start of the procedure FR-H.1) was relevant to this case. Based on an analysis of the crew data and timing, a revised timing distribution was determined (Table 20).

Table 20 - Timing Distribution - LOFW Base Case

Scenario	**Parameter**	**Briefing Hold #1 (Initial)**	**Briefing Hold #1 (Actual)**
Base Scenario	Minimum time (μ)	350	350
	Scaling Parameter (α)	200	376
	Shape Factor (β)	1.75	0.901
	Kolmogorov-Smirnov Test Statistic (K-S)	-	0.165
	Critical K-S Value (0.05 significance)	-	0.349

The revised timing distribution show a reasonable and statistically significant fit to a Weibull distribution (Figure 55). The main difference between these two distributions is that the actual crews had a longer time delay and a greater amount of variability than the timing assumed for the initial ADS-IDAC analysis. Based on the revised distribution parameters, if five timing branches were generated, the nominal branching times would be 382 seconds, 471 seconds, 603 seconds, 820 seconds, and 1440 seconds. It is important to realize that when ADS-IDAC is run, the actual branching times may not show good agreement with observed behaviors due to the available resolution of the branching times. For example, the branching time of 820 seconds does not match any crew timing particularly well. Therefore, the decision about how many timing branches to include is a tradeoff between the calculational effort and the desire to better replicate observed behavior.

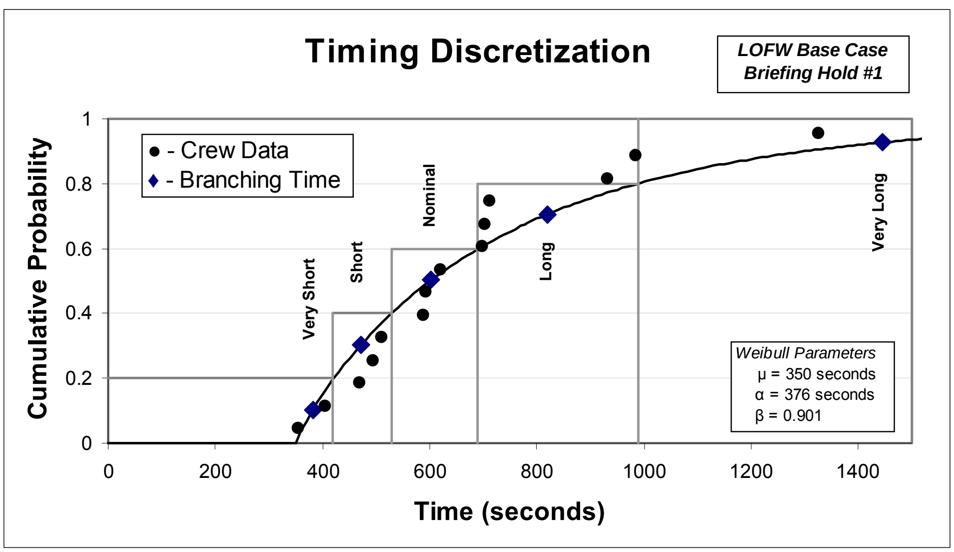


Figure 55 - LOFW Base Case Briefing Hold #1 Timing

It was initially believed that there might be significant timing variability in the delay between reaching the wide range SG level criteria that required initiation of feed and bleed and cooling and the actuation of the high pressure injection system. Briefing Hold #3 in FR-H.1 (see Section 10.3.2) was intended to model this timing variability. However, in reviewing the observed crew data, only one crew delayed initiation of high pressure injection by more than one minute once two or more wide range SG level indicators reached the 12% transition criteria. The delay time for most crews was less than 30 seconds; though the longest crew took approximately three minutes. Therefore, it was concluded that variability in the time delay between reaching conditions requiring initiation of feed and bleed cooling and the execution of operator actions to align high pressure injection was not a significant source of crew-to-crew variability for this scenario.

In reviewing the base scenario data, an unexpected source of crew-to-crew variability was identified. It was originally felt that there would be a minimal delay

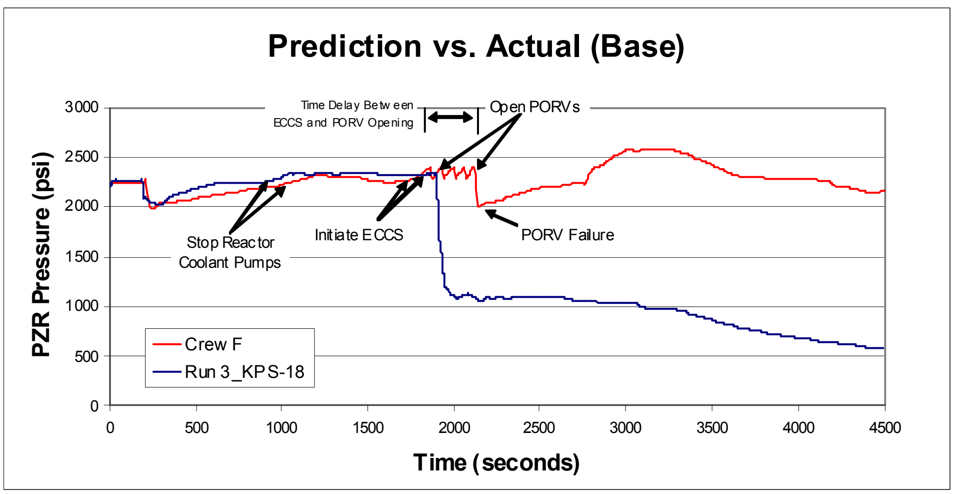


Figure 56 - Pressurizer Pressure Response (Base - Crew F)

between initiating high pressure injection for feed and bleed cooling and alignment of the pressurizer PORV bleed path. However, substantial delays between the initiation of high pressure injection and alignment of the PORV bleed path were observed for several crews. As shown in Figure 56, this delay resulted in multiple pressurizer PORV opening and closing cycles as the valve attempts to maintain pressure below the code safety relief setpoint. There were several other unexpected crew behaviors noted during the base case scenario, including a non-proceduralized action to close the main steam isolation valves.

It is not clear how a number of modeling assumptions for the nuclear plant model may affect the comparison between the Halden data and ADS-IDAC. For example, the ADS-IDAC model assumed that the condenser steam dump system remained available for a short time following loss of condensate pump flow. Additionally, since main feedwater pumps are large rotating pieces of equipment with substantial inertia, they may provide some continued feedwater flow following loss of condensate as they coast down. Because the detailed modeling assumptions used in the FRESH simulator were not available, these factors represent a source of model uncertainty for the ADS-IDAC analysis.

10.3.4.3 LOFW Complex Case Scenario Comparison to Predictions

During the complex case scenarios, Briefing Holds #1 (before the start of FR-H.1) and Briefing Hold #2 (before reactor coolant system depressurization in Step 7.a) were determined to be relevant to the data comparison. Briefing hold #3 was originally envisioned to be used prior to initiation of feed and bleed to represent a delay between the crew's decision to initiate this mode of decay heat removal and the actual execution of the required actions. Although it is likely that crews exhibited some level of variance in the time taken to initiate feed and bleed cooling, there is insufficient information in the simulator log data to support an assessment for this timing behavior. The crew data analysis for the timing of briefing holds #1 and #2 is summarized in Table 12. Similar to the results for the base case scenario, the initial

Table 21 - Timing Distribution - LOFW Complex Case

Scenario	Parameter	Briefing Hold #1 (Initial)	Briefing Hold #1 (Actual)	Briefing Hold #2 (Initial)	Briefing Hold #2 (Actual)
Base Scenario	Minimum time (μ)	350	350	140	140
	Scaling Parameter (α)	200	531	200	408
	Shape Factor (β)	1.75	1.25	1.75	1.19
	Kolmogorov-Smirnov Test Statistic (K-S)	-	0.185	-	0.104
	Critical K-S Value (0.05 significance)	-	0.361	-	0.361

timing parameter used in the ADS-IDAC analysis underestimated both the time delays and the variance among the crews. The revised timing distributions for both briefing holds show a reasonable and statistically significant fit to a Weibull distribution (see Figure 57 and Figure 58).

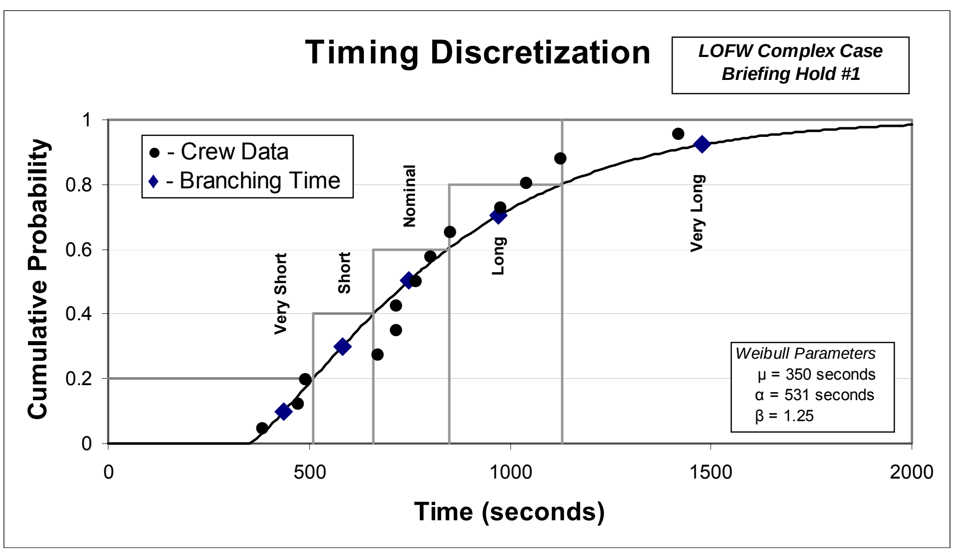


Figure 57 - LOFW Complex Case Briefing Hold #1 Timing

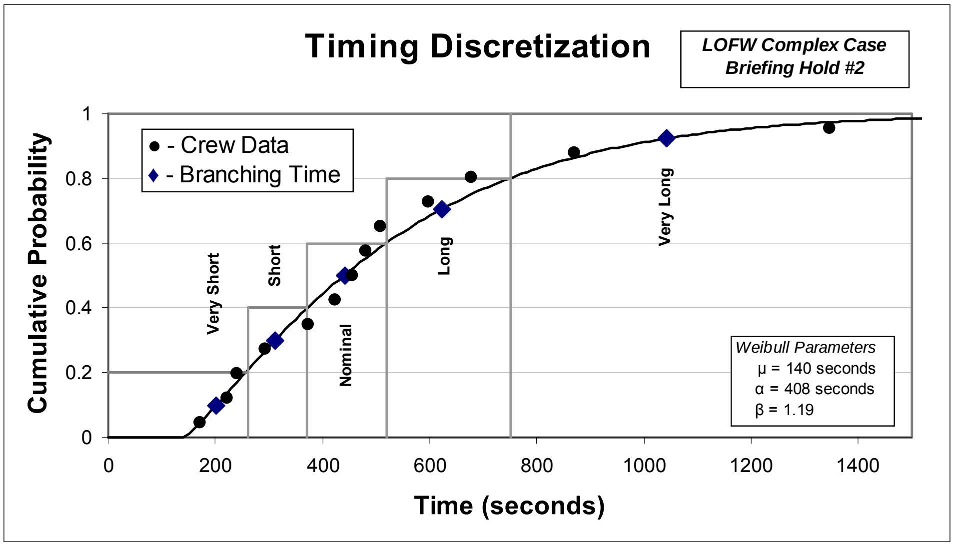


Figure 58 - LOFW Complex Case Briefing Hold #2 Timing

A significant modeling issue was identified in reviewing the complex scenario crew data. Specifically, procedure FR-H.1 specifies that the auxiliary spray system should be used to reduce RCS pressure in Step 7.a. When the ADS-IDAC model was created, it was initially believed that the pressurizer PORV could be used to fulfill the same function as the auxiliary spray valve. Therefore, no auxiliary spray system model was initially included in the ADS-IDAC nuclear plant model. However, in reviewing the FRESH simulator data, use of the auxiliary spray system generates a significantly different plant response than use of the PORV. As shown in Figure 59, the pressure response for the auxiliary spray valve is considerably slower than that for the pressurizer PORV. Additionally, the addition of coolant via the auxiliary spray system tends to increase pressurizer water level while the use of the pressurizer PORV tend to reduce pressurizer level. Consequently, if the auxiliary spray valve is

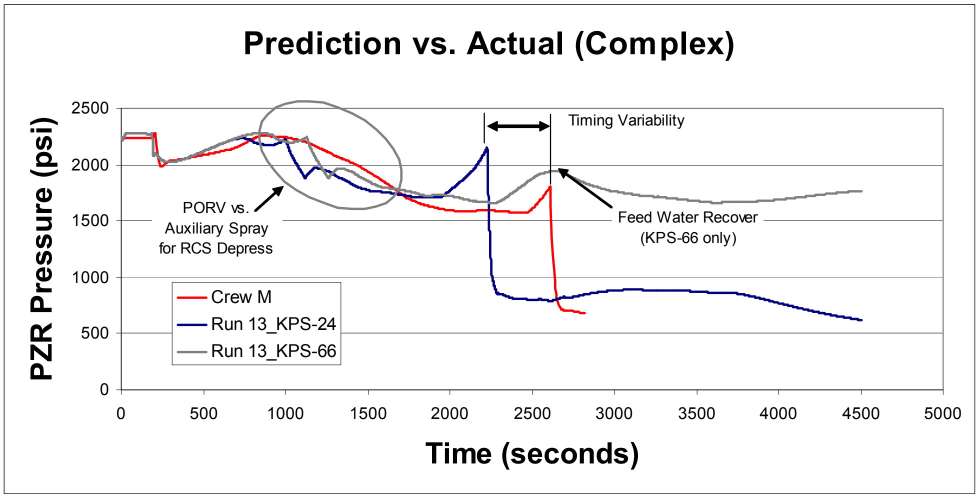


Figure 59 - Pressurizer Pressure Response (Complex - Crew M)

not included in the ADS-IDAC model, the predicted plant response will not accurately match observed behavior.

Another operator behavior that was not anticipated is operator action to maintain pressurizer pressure below the P11 permissive setpoint. Allowing pressure to increase above this setpoint would eliminate the block placed on the low pressure.

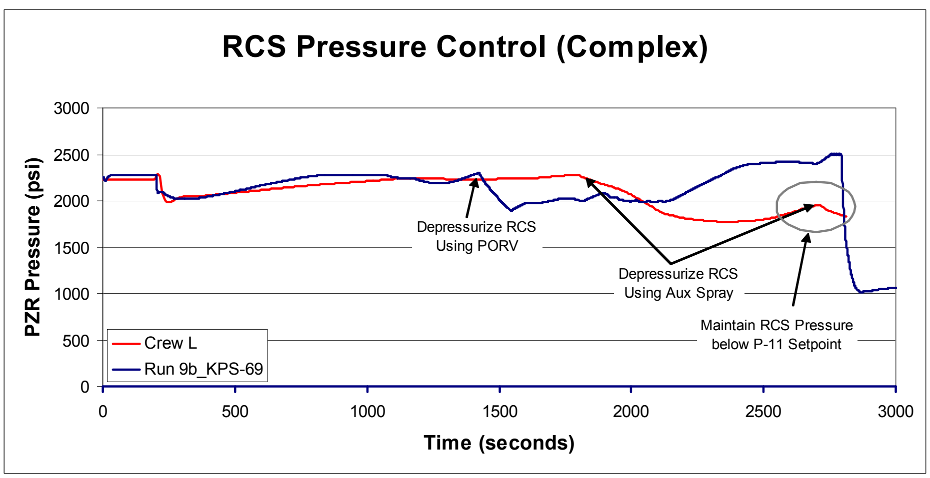


Figure 60 - Pressurizer Pressure Response (Complex - Crew L)

safety injection actuation during step 7.b of FR-H.1. Although this action is not explicitly covered by the procedures, Figure 60 shows Crew L taking action to proactively maintain pressure below the reset setpoint.

Some crews also encountered difficulty decreasing RCS pressure below the P11 permissive setpoint (approximately 2000 psi) when attempting to block the low RCS pressure safety injection actuation signal. The FRESH simulator includes a design feature that automatically shuts the pressurizer PORVs when pressure decreases below 2000 psi. Although this feature can be readily defeated by the crews during the complex scenario, some crews (e.g., Crew F) appeared to experience difficulty in doing so (see Figure 61). Because the ADS-IDAC plant model does not include this feature, it was not possible to anticipate this behavior.

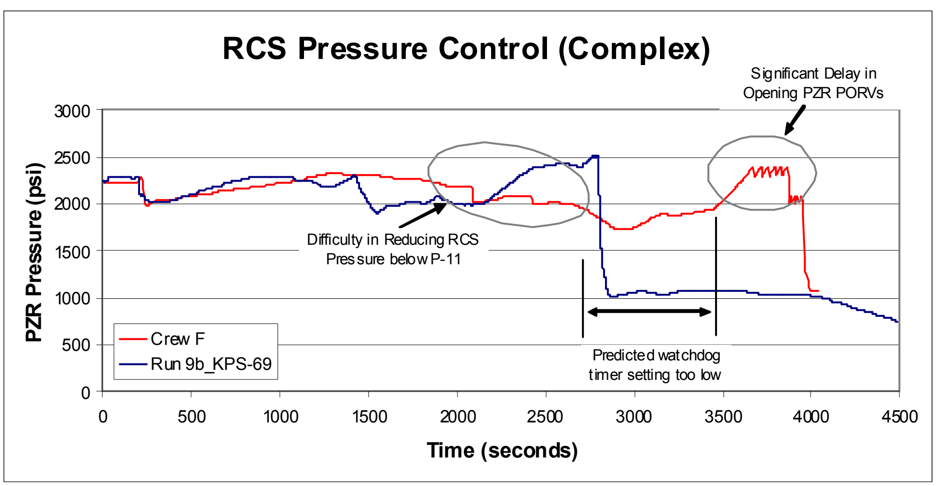


Figure 61 - Pressurizer Pressure Response (Complex - Crew F)

A final modeling issue involved the restoration of feed water flow to the SGs once secondary pressure was reduced below the condensate pump discharge pressure.

The ADS-IDAC model aligned the condensate pumps to the SGs when the appropriate pressure condition was met (see Figure 62). However, none of the Halden simulator crews aligned feedwater flow to the SGs even when the requisite conditions were met. One possible explanation is that cold feeding a hot SG can result in damage to the steam generator and place a significant thermal transient on the reactor coolant system. It is possible that the Halden crews delayed initiation of feedwater flow due to their perceived decision responsibility during the scenario. Once feed and bleed cooling is established, there is adequate heat removal from the reactor core and it is not necessary to immediately initiate cold feeding from the condensate system. Although feed and bleed cooling is not a desirable form of decay

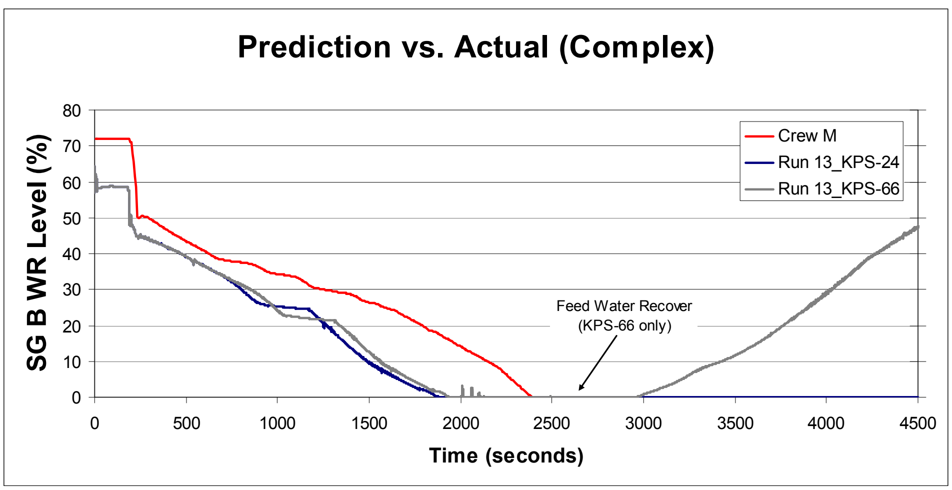


Figure 62 - SG Wide range Level Response (Complex - Crew M)

heat removal, the associated plant conditions may reduce the pressure on the crew to take immediate action to ensure the public health and safety.

10.3.4.4 Summary of Predicted and Observed Operator Behavior

A summary table of the predicted and observed operator behavior for the base and complex scenarios is provided in Appendix H, "LOFW Scenario: Crew Performance Summary." A number of key predicted performance issues were observed during the actual simulator exercises, including initiation of an early manual reactor trip, failure to trip the reactor coolant pumps in a timely manner[12], timing variabilities for key actions during the loss of heat sink procedure, failure to block certain safety injection actuation signals, and failure to establish an effective bleed path through the pressurizer PORVs. Despite this success, the ADS-IDAC model lacked several key features that were observed during the experiments. In general, these issues fell into the following categories:

- Timing calibration issues – A nominal timing distribution was used for all procedure holds in FR-H.1. Because there was very little data available to develop a more informed estimate for timing variability, the distribution parameters did not have a rigorous basis. In comparison with observed data, the assumed distributions underestimated both the average time required to execute the procedure and the variance between crews.
- Plant thermal-hydraulic model issues – A number of thermal-hydraulic modeling issues were identified during the initial comparison effort. Most significant among these issues are the failure to include the auxiliary spray

[12] Although several crews significantly delayed tripping the reactor coolant pumps during the LOFW scenarios, it was determined that these delays were due to delays in initiating procedure FR-H.1 rather than skipping the associated steps in procedure FR-H.1. Although ADS-IDAC was useful in identifying that time required to trip reactor coolant pumps could represent a significant source of crew variability, the basis for this prediction did not match actual crew behavior.

system in the thermal-hydraulic model and differences in the calibration of the wide range SG water level instrumentation.

- Procedure modeling – The observed crew performance data indicates that a number of procedure steps were performed in parallel (e.g., in several cases depressurization of the RCS continued after the safety injection P11 permissive was reached) and some of the crews did not perform certain recovery actions during the complex scenario. Although it is difficult to identify the reasons and motivations for these behaviors, it is possible that crews evaluated the time constraints and determined that there was insufficient time to effectively perform the requested actions.

Section 10.4 discusses the recalibration effort that was used to upgrade ADS-IDAC in order to better reproduce the results observed during the LOFW simulator exercises.

10.4 Recalibration of ADS-IDAC Model

The initial comparison between ADS-IDAC predictions and observed crew behaviors for the Halden LOFW scenarios showed that the ADS-IDAC approach is capable of predicting a number of contextually driven human errors. In particular, the crew variability associated with the execution of an early reactor trip and the failure to block certain automatic safety feature actuation signals were both predicted by the ADS-IDAC information driven model and observed during the empirical study. In addition, it was determined that variations in the action thresholds utilized by the operating crews to trigger key actions such as initiation of feed and bleed cooling can be readily examined with the ADS-IDAC model. However, as noted in Section 10.3.4.4, a number of limitations in the ADS-IDAC model were also identified. These issues fell into three broad categories – errors or over-simplifications in the ADS-IDAC nuclear plant thermal-hydraulic plant model, the somewhat inflexible procedure-following flow path initially used in ADS-IDAC to model the loss of heat sink procedure FR-H.1, and the lack of a number of important skill- and rule-based behaviors utilized by the crews during the scenario. In order to both improve the content of the ADS-IDAC operator knowledge base and ensure that the model is capable of realistically representing actual crew performance, the model was recalibrated based on the operating experience gained from the Halden empirical study. After the recalibration effort was completed, a final set of ADS-IDAC simulation runs was performed to verify that the model was capable of accurately reproducing actual crew behaviors using a minimal set of branching rules.

10.4.1 Thermal-Hydraulic and Operator Model Modifications

Several thermal-hydraulic, procedure, and operator modeling issues were identified during the initial comparison of ADS-IDAC predictions to observed operator behaviors. In order to resolve these issues, a number of changes were made to the RELAP nuclear plant model improve the alignment between the ADS-IDAC nuclear plant model and the Halden FRESH nuclear plant simulator. These plant model revisions included a reduction in the ADS-IDAC plant initial power level and decay heat generation, rescaling of the SG wide range level indicators, the addition of an auxiliary spray valve for use as an alternate method to reduce pressurizer pressure, and adjustment of modeling assumptions pertaining to the post trip availability of the condenser steam dump system during the base case scenario. The following specific issues were addressed in the ADS-IDAC plant model in order to better reproduce the results obtained in the FRESH simulator facility:

- *Reactor Power Level*

  The ADS-IDAC reference plant has a rated full power level of 2660 MW. This corresponds to a full power steam flow of approximately 3230 lbm/sec. In contrast, the full power steam flow for the FRESH simulator is approximately 3150 lbm/sec, or roughly 2.5% lower. Assuming that the design of the ADS-IDAC and FRESH SGs are similar, the higher power level in the ADS-IDAC model results in a quicker depletion of SG water inventory prior to the reactor trip and a higher decay heat generation rate after reactor trip. Both of these effects would cause the ADS-IDAC model to reach the SG wide range level transition criteria of 12% level before the FRESH simulator, assuming all other factors were similar. To address this issue, the initial power level of the ADS-IDAC model was reduced from approximately 100% power to 97% power. This brought the initial total steam flow for the ADS-IDAC plant model down to roughly 3155 lbm/sec, or within less than 0.2% of the FRESH simulator steam flow. In addition to the power reduction in the ADS-IDAC model, the decay heat generation rate was also slightly reduced to provide better agreement with the long term SG water level trend observed in the Halden simulator.

- *SG Wide Range Level Instrument Scaling*

Following the reduction in initial power level for the ADS-IDAC plant model, the agreement between SG wide range level response between ADS-IDAC and the FRESH facility was slightly improved but still relatively poor. The wide range steam generator level for the ADS-IDAC RELAP plant model is calculated using Equation 13:

$$WR\,Level = -\alpha + \beta\,(p_{Lower\,Tap} - p_{Upper\,Tap}) \qquad \text{Equation 13}$$

Where:

$$\alpha = \frac{\rho_g(T_{CAL})}{\left(\rho_f(T_{CAL}) - \rho_g(T_{CAL})\right)}$$

$$\beta = \frac{1}{\left(\rho_f(T_{CAL}) - \rho_g(T_{CAL})\right) g\,(h_{Upper\,Tap} - h_{Lower\,Tap})}$$

In these equations, $\rho_g(T_{CAL})$ and $\rho_f(T_{CAL})$ refer to the steam vapor and liquid density at the SG level calibration temperature T_{CAL}; g is the gravitational constant, and $h_{UpperTap}$ and $h_{LowerTap}$ are the relative heights of the upper and lower instrument taps into the SG, and $p_{UpperTap}$ and $p_{LowerTap}$ are the pressure conditions at the respective instrument tap. The calculation of SG water level requires knowledge of two plant specific parameters, namely the selected calibration temperature for the SG level instrument and the relative heights of the upper and lower instrument taps. Unfortunately, this detailed level of plant information was not available for the FRESH simulator. Therefore, the β factor in Equation 13

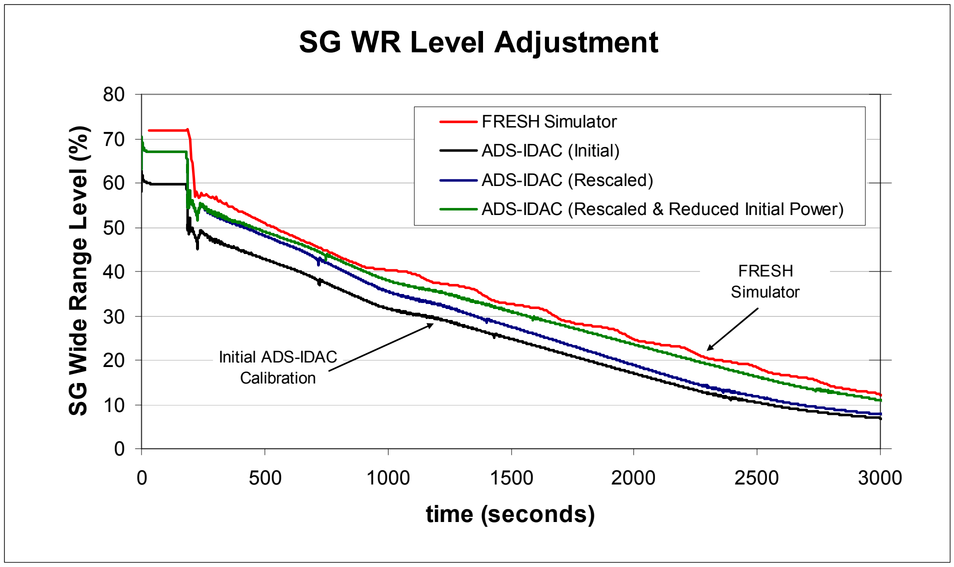


Figure 63 - Calibration of ADS-IDAC SG Wide Range Level

was adjusted to bring the ADS-IDAC wide range steam generator level into closer agreement with the FRESH simulator (see Figure 63). The net effect of this adjustment was equivalent to changing the instrument calibration temperature from 800 psia (corresponding to the normal full power SG pressure) to approximately 1000 psia (which is consistent with the normal hot shutdown SG pressure). Additionally, a lag filter was added to the RELAP level calculation to improve the stability of the instrument.

- *Auxiliary Spray*

The auxiliary spray system is used to reduce pressurizer pressure when the normal pressurizer spray system is unavailable. The driving force for the normal spray system is provided by the reactor coolant pumps. When the reactor coolant pumps are stopped (as required by FR-H.1), there is no longer a sufficient differential pressure across the spray valves to provide spray flow into the pressurizer. The auxiliary spray system directs flow from the chemical and volume control system (which is used to control pressurizer water level), into the pressurizer spray nozzle to reduce RCS pressure. During FR-H.1, the auxiliary spray system is the preferred means for pressure reduction. When the loss of heat sink procedure was initially modeled in ADS-IDAC, the analyst felt that the additional realism provided by building an auxiliary spray system model was not worth the level of effort required. Therefore, all RCS pressure reductions in the ADS-IDAC model were made with the pressurizer PORV valve. It is now clear that this approach was inadequate since the plant response to using the auxiliary spray valve is substantially different from the response obtained by using the PORV. Most importantly, opening the PORV bleeds reactor coolant from the plant, reducing pressurizer level, while the use of the auxiliary spray system adds coolant to the RCS and increases pressurizer water level. Also, the rate of pressure reduction with the spray valve is significantly slower than with use of the PORV. Consequently, the ADS-IDAC model failed to reproduce important features of the nuclear plant response during procedure FR-H.1. To remedy this situation, an auxiliary spray valve model was added to the ADS-IDAC plant. This experience served to underscore the importance of ensuring that the nuclear plant model in ADS-IDAC is as realistic as possible.

- *Condenser Availability Following Loss of Condensate Pumps*

  The normal means to remove decay heat following a reactor trip is through use of the condenser steam dump system. This system directs steam from the SGs directly into the main turbine condenser. The condenser is designed to operate at sub-atmospheric pressure conditions (i.e., under a vacuum) and steam dump flow will be isolated if condenser pressure becomes elevated. The flow from the condensate pumps supports a number of functions needed to maintain main condenser steam dumping availability, including cooling for air ejector heat exchanges and hotwell level control. Although the condenser vacuum can be maintained without condensate pump flow for a short period of time (the main heat removal function is provided by a separate circulating water system), vacuum will eventually be lost due to the build up of non-condensables or loss of hotwell level control. The availability of the condenser steam dump system following loss of condensate flow is significant for the LOFW base scenario since loss of the condenser for decay heat removal limits the options available to the operators. Because the ADS-IDAC thermal-hydraulic plant model does not have a detailed model of the main condenser, the loss of condenser vacuum following a condensate pump trip was initially simulated by blocking condenser steam dump capability 120 seconds after loss of condensate pump flow. However, in reviewing the results of the Halden simulator exercises, it appeared that the condenser remained available for a substantial period of time following loss of the condensate pumps. For better consistency with the results from the Halden simulator exercises, this time period for condenser availability following loss of the condensate pumps was increased to 3000 seconds for certain sequences. This remains a point of uncertainty in the ADS-IDAC analysis.

- *Failure of Pressurizer PORV Bleed Path*

  During the base scenario, the Halden crews experienced an additional failure in that the pressurizer PORV's bleed flowpath isolated once RCS pressure dropped below approximately 2000 psi. This condition required the operators to depressurize at least one SG in order to align a low pressure source of makeup water (e.g., firemain water). This condition was not originally included in the ADS-IDAC mode, but was easily added through the use of a conditional hardware failure branching event.

There are several differences between the ADS-IDAC plant model and the FRESH simulator that were not reconciled. A key plant difference is that the ADS-IDAC plant model includes non-return check valves on each SG main steam line while the FRESH simulator does not. The non-return valves prevent one SG from back feeding into the other two via the main steam system. The main effect of this plant difference is that opening a single SG atmospheric dump valve on the FRESH simulator permits steam flow from each steam generator to be vented to atmosphere, somewhat simplifying the SG cooldown and depressurization process. In the ADS-IDAC model, the non-return check valves block steam flow from the other SGs even if all the MSIVs are open. Additionally, some of the automatic protective functions differ between the plants. For example, the ADS-IDAC model does not include a safety feature actuation signal on high differential pressure between the SGs while the FRESH simulator does include this feature. These differences can limit the ability of ADS-IDAC to accurately reproduce contextual factors that may influence operator behaviors when these features are important. For example, one crew during the Halden empirical study actuated the emergency core cooling system during SG depressurization due to a high differential pressure between SGs. This inadvertent action served to accomplish a key feature of FR-H.1, namely alignment of high pressure feed flow to the reactor coolant system. However, this feature is not included in the design for the ADS-IDAC reference plant and thus could neither be predicted nor simulated. This serves to underscore the need to accurately represent the real plant model when using a simulation tool such as ADS-IDAC to predict operator behaviors (or at least appreciate the limitations of such an approach).

In addition to nuclear plant modeling changes, a number of issues regarding the implementation of the loss of heat sink procedure (FR-H.1) were also noted during the comparisons with the Halden crews. Most notably, the Halden crews appeared to perform a number of actions in parallel. For example, during the complex scenario, step 7.a of FR-H.1 directs the operators to reduce RCS pressure below approximately 2000 psia and then block certain safety feature actuation signals per step 7.b. It was initially assumed that crews would terminate the depressurization of the RCS shortly after achieving the required RCS pressure condition needed to block the low pressure safety injection actuation signal. However, some crews continued depressurization of the RCS for a substantial period of time after completing the actions of FR-H.1 Step 7.b. In some cases, the crews continued forward in the procedure with additional actions, such as depressurization of the SGs, with RCS depressurization still in progress. Other crews continued depressurization of the SGs well beyond the point at which low pressure condensate could be used to supply makeup flow. These issues indicate that crews exhibited significant variations in their interpretation of key procedure steps, often leading to different plant states and situational contexts. In order to reproduce these types of behaviors, the following revisions were made to the ADS-IDAC loss of heat sink procedure model:

- *Additional Watchdog Timer*

  In order to permit the modeling of time dependent actions within ADS-IDAC, the model includes "watchdog timers" that can be set and read by the operators. These timers are useful for initiating actions or behaviors that are executed after a fixed time delay, such as initiating certain actions within a fixed time after commencing a procedure or switching to an alternate means to accomplish a function after a predetermined interval. For example, in the original ADS-IDAC FR-H.1 model, a watchdog timer was used during the complex scenarios to initiate the transition to feed and bleed cooling since the SG wide range level instruments could not be used due to biasing factors applied to two of the indictors. The timer was initialized when FR-H.1 was started and periodically checked as the operator progressed through the procedure. When the timer reached a threshold value, the operator initiated action to establish feed and bleed cooling. In reviewing the Halden crew data, it was noted that during the complex scenario several crews attempted to depressurize the RCS using auxiliary spray, but then switched to the pressurizer PORV after a time delay. This switching of depressurization method might have been motivated by a belief that RCS depressurization using the auxiliary spray valve was either too slow or ineffective. In order to model this type of behavior, an additional watchdog timer was added to the ADS-IDAC model that triggers a switch to the pressurizer PORV if the auxiliary spray system does not effectively lead to the desired plant conditions before the timer expires.

- *Time Availability for Performing Key Actions*

  Some crews during the complex scenario appeared to skip procedure actions associated with depressurization of the RCS or the SGs. This may have occurred if the crew believed that insufficient time was available to execute the required action. Under these circumstances, a crew may elect to maintain stable plant conditions and initiate feed and bleed cooling to ensure adequate decay heat removal prior to attempting further recovery actions. This behavior is a form of step-skipping behavior in that the crew elects to deviate from a procedure objective in order to achieve a goal considered to be more relevant to the perceived situational context. Although a future version of ADS-IDAC may include sufficient modeling features to handle this type of procedure step skipping, the current version lacks this capability. Therefore, in order to model this crew behavior, the procedure was modified to verify the remaining time of the watchdog timer used to trigger the transition to feed and bleed initiation. If the remaining time was less than a preset threshold, the associated action was skipped. The time threshold could be viewed as a crew preference or tendency that characterizes the willingness of the crew to initiate a fairly complex recovery action with limited time.

- *Blocking Cold Feed of the Steam Generators*

  It was determined that none of the Halden crews initiated cold feeding of the steam generators during the complex scenario, even when plant conditions would support such an action. Since one of the main objectives of FR-H.1 is to reduce SG pressure below the condensate pump discharge pressure so that the condensate system can be used as an alternate source of SG makeup water, this observation was somewhat surprising. However, cold feeding a steam generator has the potential to damage plant equipment and may lead to a significant transient on the reactor plant. Consequently, the crews may have felt that the decision to initiate cold feeding was beyond their level of responsibility or should only be done in consultation with more senior plant management staff. Given that the establishment of stable feed and bleed cooling provides adequate core cooling, initiation of cold feeding does not necessarily need to be done in a rapid manner. In order to model this behavior in ADS-IDAC, a crew preference to delay cold feeding a SG was added to the operator knowledge base. If this preference is activated, the ADS-IDAC procedure model blocks initiation of cold feeding during FR-H.1.

- *Establishing the Pressurizer PORV Relief Path*

  During the base scenario, a plant equipment failure that was not initially described in the empirical study description was used during the actual crew experiments. Specifically, once the operators aligned a bleed path by opening one or more pressurizer PORV, and the RCS pressure decreased below approximately 2000 psi, the PORV bleed path would isolate. Once the bleed path isolated, the operator would be unable to reopen the valves. This condition was not included in the initial ADS-IDAC procedure model since this failure was not anticipated. However, in this situation, FR-H.1 step 16 directs the operators to depressurize at least one SG in order to align a source of low pressure makeup water. In order to better match Halden crew performance with ADS-IDAC, the appropriate steps to depressurize the SGs were added to the procedure model to support these actions. It was noted that several crews had significant delays in aligning the pressurizer PORV bleed path following initiation of high pressure coolant injection. Although the time delay between initiation of injection flow and opening the PORV valves was only approximately two to three minutes for most crews, several crews experienced delay of more than five to ten minutes. This behavior was easily captured in the ADS-IDAC model by increasing the time required to open the pressurizer PORVs.

The revised FR-H.1 procedure model based on these enhancements is shown in Figure 64. The changes to the procedure model relative to the initial version (Figure 50) are highlighted. The increased flexibility contained in the revised implementation model for FR-H.1 improves the capability of ADS-IDAC to reproduce observed operator behaviors.

Summary of ADS-IDAC Model Recalibration

In summary, the final recalibration focus area involved the development of additional mental beliefs to enhance the operator knowledge base in ADS-IDAC. The creation of new mental beliefs was largely driven by the desire to better model observed operator behaviors during the execution of FR-H.1. These behaviors include skipping of certain procedure sections, blocking the addition of cold feedwater to the SGs, performing certain actions in parallel, and taking actions to protect plant equipment or maintain key plant parameters within a specified control band. Specifically, the following mental beliefs were developed:

- Equipment Protection - A mental belief to activate a rule-based action to trip all reactor coolant pumps when a loss of subcooling margin occurs following initiation of emergency core cooling.

- Crew Preferences - A mental belief to characterize the crew's preference for alignment of a cold feedwater source to a hot and dry steam generator. Based on a review of the Halden data, even when crews had established the required plant conditions to align the condensate system to provide SG makeup, this action was not performed. Therefore, the decision to delay initiation cold feeding to the SGs was modeled as a crew preference. Similarly, apparent crew preferences to skip

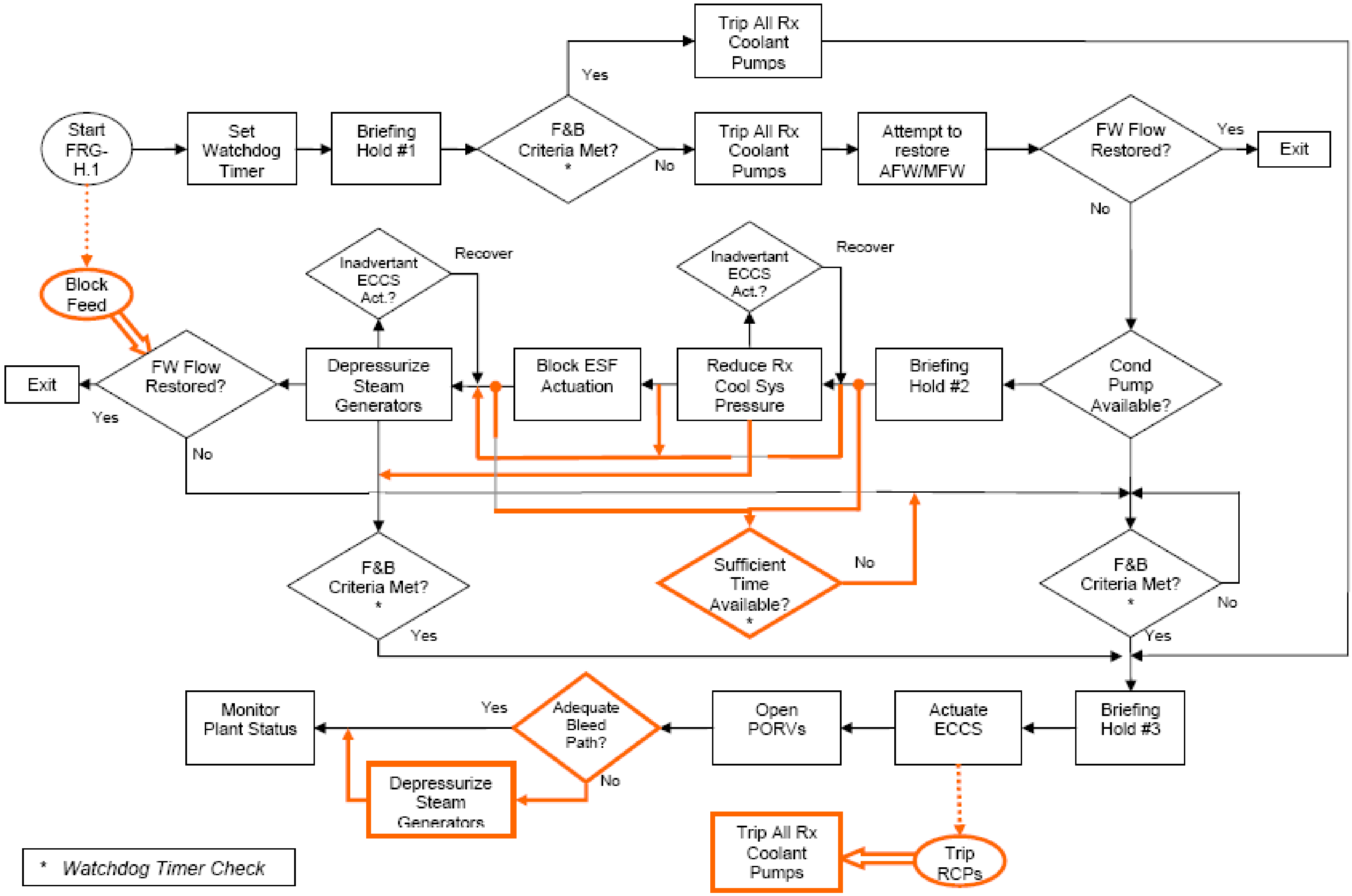


Figure 64 - Revised Loss of Feedwater Procedure Model

RCS and/or SG depressurization during the complex case were also modeled with mental beliefs.

- Performance of Parallel Actions – During the complex scenario, it was noted that some crews performed some FR-H.1 steps in parallel. For example, crews continued the RCS depressurization directed by step 7.a even after the desired safety feature blocking signals had been actuated in step 7.b. Furthermore, some crews initiated the SG depressurization directed by step 7.c without first terminating the RCS depressurization initiated in step 7.a. To model these types of parallel procedural actions, additional mental beliefs to shut the auxiliary spray valve, secure SG depressurization, and stop the reactor coolant pumps were added to the operators' skill- and rule-based action queue. The time delay for performing each of these actions was established to better represent actual crew behavior. Using this model, the ADS-IDAC crew could then continue the execution of procedure FR-H.1 while executing a previous procedure step in parallel.

- Maintaining Plant Parameters Within a Control Band – During the complex scenario, an inadvertent safety injection would lead to a main feedwater system automatic isolation and further complicate plant recovery. Therefore, it is important to avoid inadvertent actuation of a safety injection during depressurization of the SGs. Consequently, step 7.b. of FR-H.1 directs the operators to block certain safety injection signals once RCS pressure is reduced below the blocking permissive setpoint of approximately 2000 psi. Because of the design of the blocking control system, if RCS pressure increases above 2000 psi, the safety injection signals are unblocked. Although not explicitly stated in the procedure, once a crew successfully blocks these signals, RCS pressure must be maintained below the permissive setpoint. In order to model this behavior, a mental belief was added to the operator knowledge base to use the auxiliary spray system as needed to maintain pressure below the permissive setpoint once the safety injection signals were blocked.

Once these modeling changes were added to the operator knowledge base and nuclear plant model, ADS-IDAC was run to determine if the actual crew behaviors observed during the Halden exercises could be reproduced. Although this does not serve to validate the predictive capability of the ADS-IDAC model, it provided useful information about the modeling capabilities of ADS-IDAC and the feasibility of using a simulation approach to predict operator behavior. Additionally, this effort served to significantly extend the scope and level of detail of the operator knowledge base. The results of the final comparison effort are discussed in Section 10.4.2.

10.4.2 Final Comparison to HAMMLAB Experimental Results

In order to verify that the modeling improvements successfully reproduced the actual observed crew behaviors during the Halden LOFW scenarios, a final set of ADS-IDAC runs were performed to verify the revised ADS-IDAC model. The objective of these runs were twofold: first to verify that the enhanced ADS-IDAC plant model and knowledge base adequately reflected actual crew performance, and second to ensure that the range of crew-to-crew variability observed during the experiments could be reproduced with a minimal set of branching rules.

10.4.2.1 LOFW Base Case – Final Comparison

The main source of observed crew-to-crew variability for the base case scenario involved the decision to trip the reactor early (i.e., prior to automatic reactor trip) and the timing of key operator actions such as tripping the reactor coolant pumps and opening the pressurizer PORV bleed path. It was determined that seven branching rules were adequate to represent the range of crew-to-crew variations observed during the base case scenario. The specific branching rules involve the following elements:

- Early detection of the loss of feedwater condition leading to a manual reactor trip. The early manual reactor trip was initiated by activation of a mental belief associated with the occurrence of a loss of feedwater event. Two branches were used to represent crew variability (either the crew recognized the loss of feedwater early and manually tripped the reactor, or the reactor was automatically tripped due to low SG water level).

- The delay time between initiation of procedure FR-H.1 and initiation of feed and bleed cooling. As discussed in Section 10.3.2, a watch dog timer is used to force an early transition to feed and bleed cooling. It was determined that the feed and bleed initiation threshold for all fourteen base case crews could be modeled with four watchdog timer branches. Three branches represent an early transition to feed and bleed, while the fourth timing branch is set sufficiently high to permit wide range SG level to drive the crew transition.

- The time delay between trip of the reactor and stopping of the reactor coolant pumps (Briefing Hold #1). It was determined that generating five timing branches on briefing hold #1 were sufficient to cover crew variability.

- The time delay between initiation of high pressure injection for feed and bleed cooling and opening of the pressurizer PORV. Three timing branches for opening of the PORV adequately represented crew variability for the base case.

- For those crews that recognized the occurrence of the loss of feedwater event and initiated a pre-emptive manual reactor trip, the time to execute a manual reactor trip was included as a branching point. Two timing branches were sufficient to represent crew-to-crew variability.

- The time delay between reaching the SG wide range level condition requiring initiation of feed and bleed cooling and initiation of high pressure injection. For all but one crew, this time delay was negligible.

- The time to initiation of depressurization of a SG once it was determined that the PORV bleed path was blocked. This was adequately modeled with two timing branches.

The failure to trip all three reactor coolant pumps relatively early in the scenario by one crew (Crew L) during the base case scenario was handled through the use of a conditional run where the tripping of the loop 1 reactor coolant pump was blocked during execution of FR-H.1 Step 3. Although ADS-IDAC has the capability to explicitly model the tripping of only two pumps, a significant number of sequences must generated to obtain the proper combination of tripped and untripped pumps. To avoid sequence explosion, a conditional run was used to avoid the generation of excessive branch sequences. Although this approach adequately models the plant response observed for Crew L (see Figure 85 and Figure 86 in Appendix I), it does not realistically model actual crew behavior. Based on additional feedback from the Halden Reactor Project staff, it was learned that this crew tripped two reactor coolant pumps prior to starting FR-H.1 to minimize heat input to the RCS. Although it took the crew a significant amount of time to transfer to procedure FR-H.1, they tripped the third pump in accordance with procedure direction. Therefore, the crew did not actually skip or miss the step in FR-H.1. However, this crew behavior could be captured within ADS-IDAC through the creation of an additional rule-based action to trip two reactor coolant pumps when a loss of secondary heat sink condition is perceived and the assignment of a long delay time for initiating FR-H.1. Because the simulator log data does not indicate when a crew actually starts a procedure, this issue highlights the need to augment quantitative experiment data with qualitative observations.

Ideally, a single ADS-IDAC simulation run could be used to exercise these seven branching rules and generate specific sequences that correspond to each of the Halden crews. Unfortunately, even with this relatively small number of branching rules, more than one thousand separate sequence branches would need to be generated. Because the underlying plant thermal-hydraulic model runs just slightly faster than real time (one hour of simulated time takes just slightly less than one hour of actual time), generating the dynamic event tree for this case on a single computer could take in excess of 30 days. Therefore, in order to make the generation of the dynamic event tree manageable in a reasonable amount of time, a series of conditional runs, each exercising only three or four branching events with all other conditions held to a nominal condition, were performed. This allowed efficient generation of sequences representing each of the Halden crews using the above branching rules without wasting computation effort on sequences that were not observed. The results of these conditional runs are provided in Appendix I. The agreement between the ADS-IDAC generated sequences using the above branching rules and the actual observed crew behaviors was reasonable, capturing most major sources of crew-to-crew variability.

10.4.2.2 LOFW Complex Case – Final Comparison

The complex case scenarios involved a significantly greater amount of crew-to-crew variability than the base case. This is not unexpected since the complex case required a greater number of operator actions in order to partially depressurize the RCS, block safety injection actuation signals, and attempt to depressurize at least one SG to align low pressure feedwater makeup capability. As in the base case, the main source of crew-to-crew variability was the timing of key operator actions. For the complex case, these key actions involved tripping the reactor coolant pumps (Briefing Hold #1), initiation of RCS depressurization (Briefing Hold #2), transition to feed and bleed cooling, opening of the pressurizer PORV to establish a bleed path, and initiation of SG depressurization. However, in addition to timing variability, there were also several examples of control input variabilities observed in the Halden crews during the complex scenarios. In particular, crews exhibited variability in the control of both the steam dump control system and operation of the auxiliary spray and pressurizer PORV valves during RCS depressurization. In general, ten branching rules captured the major sources of crew-to-crew variability in the complex scenario. The specific branching rules included the following elements:

- The delay time between initiation of procedure FR-H.1 and initiation of feed and bleed cooling. As discussed in Section 10.3.2, a watchdog timer is used to force an early transition to feed and bleed cooling. It was determined that the feed and bleed initiation threshold for all fourteen base case crews could be adequately modeled with six watchdog timer branches. During the complex scenario, the biasing of the wide range SG level indicators resulted in the watchdog timer being the sole means to initiate the transition to feed and bleed cooling. The quickest transition from start of FR-H.1 to initiation of feed and bleed cooling was approximately five minutes while the longest was more fifty minutes. Most crews fell in the range of 25 to 45 minutes.

- The time delay between trip of the reactor and stopping of the reactor coolant pumps (Briefing Hold #1). It was determined that generating five timing branches on Briefing Hold #1 were sufficient to cover crew variability.

- The time delay between tripping of the reactor coolant pumps and initiation of RCS depressurization in accordance with step 7 of FR-H.1 (Briefing Hold #2). It was determined that generating five timing branches on Briefing Hold #2 were sufficient to cover crew variability.

- The amount of time the crew used the auxiliary spray valve to depressurize the RCS prior to switching to the pressurizer PORV to complete the depressurization (Watchdog Timer #2). Eight crews exclusively used the auxiliary spray valve to depressurize the RCS, one crew used only the pressurizer PORV to depressurize, and three crews switched from the auxiliary spray valve to the PORV after a time delay. One crew did not appear to attempt any means to depressurize the RCS. To model these variations, two branches were generated on the setting of watchdog timer #2 – a nominal value and a value that would force an early transition.

- For the crews that utilized the auxiliary spray valve to depressurize RCS, there was some variability in the rate of depressurization established by the crew. Three control settings were used to model this variability – a nominal control setting, a reduced control setting, and a closed setting (to model the one crew that did not use auxiliary spray).

- When the crews resorted to use of the pressurizer PORV to depressurize the RCS during step 7 of FR-H.1, they appeared to use a more restrictive pressure threshold to terminate the RCS pressure decrease than when the auxiliary spray valve was used. This may have been motivated by the relatively rapid pressure decrease associated with use of the PORV and the desire to avoid an inadvertent safety injection actuation. To model this behavior in ADS-IDAC, the procedure expectation that initiates closure of the PORV was adjusted to approximately 2100 psi rather than below the P11 permissive setpoint (less than 2000 psia).

- Once a crew reduced RCS pressure below the P11 permissive setpoint, the low pressure safety injection signal could be blocked. However, if pressure increased above the P11 permissive, the safety injection signal would automatically unblock and re-enable the safety injection. During the LOFW scenario, the tendency is for RCS pressure to increase; therefore, if the crew delayed depressurization of the

SGs, RCS pressure could increase above the P11 permissive. Consequently, some crews operated the auxiliary spray valve to maintain RCS pressure below the P11 setpoint. This was modeled by generating a binary branching event on a mental belief that activated a goal to keep RCS pressure below the permissive setpoint.

- The time delay between initiation of high pressure injection for feed and bleed cooling and opening of the pressurizer PORV. Three timing branches for opening of the PORV adequately represented crew variability for the complex case.

- Control input for the steam dump valve during SG depressurization (i.e., the rate of SG depressurization) and the time delay in initiating depressurization. Variations included nominal, slow, and fast depressurization.

- The time delay between initiation of high pressure injection for feed and bleed cooling and opening of the pressurizer PORV. Since the majority crews had a minimal time delay between initiation of high pressure injection and opening of the PORV, only two branching events were used to model this variability – a nominal condition and a long delay of approximately 10 minutes.

Because of the number of branching rules and the associated number of sequences generated by each rule, it was not possible to reproduce all crew behaviors with a single ADS-IDAC simulation run. Similar to what was done during the final base case comparisons, a number of conditional runs, each with three to four branching rules, were run to model each Halden crew. This allowed the branching rules to be fully exercised without the generation of an excessive number of branching rule permutations and the associated sequence explosion. The results of these conditional runs are provided in Appendix J. The agreement between the ADS-IDAC generated sequences using the above branching rules and the actual observed behaviors was reasonable following recalibration.

10.5 Validation Conclusions

Although the initial set of ADS-IDAC predictions for the LOFW scenario highlighted a number of limitations of the model, the recalibration effort resulted in significant improvement to the ADS-IDAC plant model and operator knowledge base. This experience has demonstrated that ADS-IDAC simulation code provides a flexible framework for data collection and analysis. In particular, the pace and timing of procedures, interpretation of procedure steps, and execution of non-proceduralized actions can all be captured within the ADS-IDAC knowledge base framework. More importantly, these changes can be incorporated into ADS-IDAC without changing the underlying computer code. This provides a level of confidence that the ADS-IDAC modeling structure is robust and adaptable.

The final post-calibration comparisons between the ADS-IDAC results and observed behaviors during the Halden exercises illustrate that ADS-IDAC is capable of realistically modeling both nuclear plant response and operator behaviors. This experience also served to underscore the importance of ensuring that the thermal-hydraulic plant model accurately represents the physical plant system and includes all significant plant controls, indications, and alarms in order to adequately model contextual factors.

11. Conclusions

11.1 General Conclusions

As a result of this research project, the capabilities of the ADS-IDAC simulation model have been dramatically improved. Improvements include an increased capability to realistically represent operator knowledge, skills, and problem-solving styles, and implementation of dynamic performance influencing factors which reinforce the man-machine feedback loop and strengthen the transient modeling capabilities of ADS-IDAC. Complex operator mental models of plant behavior that guide crew actions can now be represented within the ADS-IDAC mental belief framework. Branching rules can now be created to simulate slow or fast procedure execution speed, skipping of procedure steps, reliance on memorized information, activation of mental beliefs, variations in control inputs, and equipment failures. The implementation of a plant functional decomposition and diagnostic engine strengthened the ability to model knowledge-based actions and other cognitive feature such as procedure step-skipping. From this study, the following has been learned:

- A relatively small number of branching rules are capable of capturing a wide spectrum of crew-to-crew variabilities. Therefore, real operator decisions and behaviors can be efficiently modeled within the ADS-IDAC framework.
- Compared to traditional static risk assessment methods, ADS-IDAC can provide a more realistic assessment of human error events by directly

determining the effect of operator behaviors on plant thermal hydraulic parameters. This shifts the analysis from an assessment of isolated operator actions to a more holistic assessment of the control room situational context and the integrated impact of a spectrum of possible operator actions on the reactor plant.

- Because model-based HRA techniques such as ADS-IDAC attempt to capture underlying cognitive processes that drive crew behaviors, these models provide an efficient framework for capturing actual operator performance data such as timing of operator actions, mental models, and decision-making activities.

Taken together, these factors improve the ability of ADS-IDAC to model complex crew behaviors and bring the state-of-the-art in human reliability assessment closer to predicting well-intentioned but deleterious knowledge-based actions that can lead to significant accident events.

11.1.1 Implementation of the IDAC Cognitive Model

This research effort has resulted in significant improvements in the implementation of the IDAC model within a dynamic probabilistic risk assessment environment. This project has enhanced the modeling of each cognitive phase of the IDAC model in addition to improving the modeling of the operator’s mental state through implementation of static and dynamic performance influencing factors. In particular, the development of a practical plant functional decomposition has linked information processing, decision-making, and action execution in a coherent, information-driven framework. The functional decomposition represents an operator's mental model of the nuclear power plant by linking each plant control, indicator, and alarm to the function it serves. By connecting the plant functional decomposition to the diagnostic engine, the operator's situational assessment can be used to influence information processing and action execution. This has dramatically improved the realism of the control panel scanning model by driving the information collection process to the areas of highest perceived need. Furthermore, by connecting the relevance of potential actions to the operator's dynamic assessment of the plant status, a greater level of realism can be applied to modeling errors of omission with the procedure step-skipping module. Collectively these factors have enhanced the ability of the ADS-IDAC approach to incorporate the influence of information processing and contextual factors into the prediction of human error events.

11.1.2 Crew Variability Modeling and Predictive Capability

A basic assumption of the ADS-IDAC approach to human reliability analysis is that crew deviations from normative/expected behavior highlight potential human error events. By better understanding the factors that cause deviations leading to a degraded plant state, the prediction of error events can be improved. A key advantage of a dynamic simulation approach is that the impact of crew deviations on the plant state can be explicitly determined and fed back to the operator. This further enhances the rich contextual information available with this modeling approach. ADS-IDAC now has enhanced capabilities to model crew-to-crew variabilities arising from information processing, decision-making, and action execution. Although the branching rule set can be used to represent a diverse spectrum of potential operator actions, it is also essential that realistic operator behaviors can be modeled with a minimal number of branching rules. If an excessive number of branching options were required to accurately represent observed crew behavior, the computational challenge associated with sequence explosion would reduce the effectiveness of ADS-IDAC as an analysis tool. Fortunately, the calibration and validation effort of this study demonstrated that a relatively small number of branching options can adequately represent a wide range of crew-to-crew variability.

11.1.4 Data Collection and Management

A significant challenge for all human reliability analysis methods is the collection and analysis of human performance data which is often not available in a form that can be readily adapted to many HRA approaches. The ADS-IDAC knowledge base addresses this challenge by providing a flexible and expandable framework for data collection. As demonstrated in this project, data obtained from observations of actual control room operators, such as timing of actions and interpretation of procedural requirements, can be readily adapted to the knowledge base framework. An additional strength of ADS-IDAC is that mental models used by the operators to diagnose and predict plant system responses and drive the decision-making process can be assimilated into the data framework.

11.2 Future Work

Although this project successfully demonstrated the feasibility of using a dynamic simulation-based approach to human reliability analysis, it also highlighted numerous areas where the method could be improved. In general, future development efforts should concentrate on continued enhancement of the IDAC model implementation and validation. Additionally, improved methodologies for integrating the ADS-IDAC approach into a coherent framework for the prediction of human error events should be developed. Specific areas for future work include cognitive model enhancements, improved modeling of the control room crew environment, improved dependency modeling through the automated generation of branching events, improved accident sequence probability quantification, development of importance measures to better communicate analysis results, capability for modeling post-core damage scenarios, and additional validation.

11.2.1 Cognitive Model Enhancements

Although this research project has improved the implementation of the operator cognitive model, particularly through the development of the functional decomposition and diagnostic module, further work needs to be done. The current method of accomplishing information filtration is largely by shifting the operator's attention focus through the use of the control panel scanning model. The information biasing filters are statically assigned prior to a simulation run without any additional provision for dynamic adjustment of information bias during an accident scenario.

Further development could improve the information filtering process through a more direct linkage to the operator's situational assessment. This linkage could improve the filtering process by connecting passively and actively gathered information to the salience of the information for the operator (similar to how the procedure step-skipping model handles the salience of actions involving a target component). Another needed enhancement is better handling of dependencies. For example, the current step-skipping model treats every action as independent. This can result in the generation of excessive numbers of branching events when a set of similar actions might be skipped by the operator. For example, when a procedure step directs the crew to trip all reactor coolant pumps, each pump is treated independently. In reality, it is highly likely that the operator would either successfully trip all the pumps or not trip any pumps. Less likely would be various permutations of tripping only one or two pumps. In order to better handle these types of situations, a dependency model that includes consideration of recent actions (and recently skipped actions) could be developed. A stronger dependency modeling would provide a more realistic simulation capability and reduce computational burden by focusing effort on most probable scenarios. Related to this concept of improving the grouping of related actions, the procedure following model could be improved to handle a hybrid procedure-following/knowledge-based problem solving strategy. This would permit the modeling of crew behaviors to skip entire sections of a procedure that were deemed to be non-relevant and jump to either a procedural or knowledge based action that more directly address a perceived plant need. Finally, the cognitive model could be enhanced by adjusting the currently static thresholds for detecting an abnormal condition and procedure step-skipping based on the operator's perceived information and mental state. This would further enhance the model's level of realism and improve the computational efficiency of ADS-IDAC by reducing the need to perform multiple simulation runs to explore variations in threshold preferences.

11.2.2 Crew Information Sharing and Resource Management

Although ADS-IDAC is currently based on a crew model framework, the operators (a decision-maker and an action-taker) operate in a relatively rigid environment. Additionally, there is only limited sharing of information among the crew members. An improved model would enhance communication and allow more sharing of perceived information. Additionally, the crew model would be more realistic with additional crew members such as another action taker (to model a second reactor operator) and a consultant (to model a shift technical advisor). The enhanced crew model would also allow ADS-IDAC to be used to explore issues associated with operator task allocation, resolution of differences in perceived information among crew members, and additional recovery potential when additional operator resources are available.

11.2.3 Automated Branch Generation

The current version of ADS-IDAC requires the analyst to pre-designate a set of branching rules for a given scenario. The exact topology of the resulting event tree is not known by the analyst, but the scope of event tree is clearly limited by the analyst. If an adequate range of branching options is not selected, important crew deviations and human error events may be missed. For example, static performance influencing factors such as goal and strategy selection tendencies, thresholds for diagnosing accident conditions, and handling of memorized information are pre-designated before simulation. Similarly, the analyst must decide on the number of timing branches that will be generated during the simulation and the specific branching probabilities for each knowledge-based mental belief. Because of computational limitations, it is not currently possible to explore every possible combination of each branching event. Therefore, the analyst must limit the scope of the analysis by selecting a limited set of branching rules. This process could be improved if a higher tier of executive control rules could be implemented that would dynamically assess the need to generate branching events during a simulation. For example, contextual factors that lead to large crew-to-crew variations could be associated with the generation of a large number of performance related branching rules. Conversely, contextual factors that lead to relatively consistent crew performance would be associated with limited branch generation. In this manner, ADS-IDAC could drive the dynamic event tree toward more interesting scenarios without wasting computational effort on scenarios unlikely to identify potential human error events.

11.2.4 Improved Accident Sequence Probability Quantification

Although certain stochastic behaviors, such as the timing of operator actions, are rigorously modeled in the current version of ADS-IDAC, the probabilistic quantification of accident sequences was not an objective of this work. Instead, this research focused on qualitative modeling of operator behaviors. To the extent possible, placeholders have been left throughout the simulation model to support future quantification capabilities, but the probabilistic models within ADS-IDAC require further development and validation.

11.2.5 Post Processing and Communication of Analysis Results

Dynamic probabilistic analysis methods generate significant amounts of data. In addition to information that is common to most risk assessment methods such as hardware failures, operator actions, and system end states, a dynamic simulation model also generates large amounts of rich contextual information such as the time history of plant parameters, alarm histories, and factors that influence the decision-making process. At the present time, methods to produce importance measures and other simplified means to communicate analysis results are not well developed. Consequently, a future work should address the need for dynamic probabilistic risk assessment importance measures and improved methods to illustrate and communicate analysis results.

11.2.6 Severe Accident Capability

Severe accidents refer to scenarios that result in a core damage event. The current thermal-hydraulic engine for ADS-IDAC, RELAP5, is not capable of modeling core damage progression and fuel melting. This places a significant limitation on the model for assessing the radiation release and public health consequences arising from a human error event. Future work could focus on improving the plant modeling capability of ADS-IDAC by linking the analysis tool to a thermal-hydraulic engine capable of modeling core damage scenarios such as the MELCOR code (similar to the approach used by ADAPT-MELCOR and MC-DET). Another option would be to link ADS-IDAC to a code suite such as SCDAP/RELAP which would preserve the existing RELAP plant model while adding capability to simulate fuel heat up and oxidation, fuel melting, and core relocation.

11.2.7 Further Model Validation

The current research study has only performed an extremely limited review and validation of the ADS-IDAC model. An important area for future work is the continued validation of the ADS-IDAC model, including the development of an acceptable approach to verify the quantification of performance influencing factors. Additionally, there are many areas of the current model where simplified estimations and judgment were used to provide needed data. For example, a simplified screening approach was used to populate the relationship data used to link perceived symptoms to individual event diagnoses. Similarly, the plant functional decomposition was based on an informed estimation rather than a true expert elicitation process. The current research effort was focused on demonstrating the feasibility of a realistic ADS-IDAC modeling approach to human error prediction. Since this study has successfully demonstrated the feasibility of a dynamic simulation approach for human performance modeling, future work should include refinement and validation of data that support the underlying cognitive model. It should also be noted the ADS-IDAC has only been exercised for two accident scenarios, a steam generator tube rupture and a loss of feedwater event. Further validation studies could focus on expanding the portfolio of analyzed events to provide a higher level of confidence that ADS-IDAC can handle a wide array of accidents. Finally, thermal-hydraulic models of additional plant types (e.g., different types of pressurized water reactors, boiling water reactors, and new and advanced reactor designs) should be developed to further expand the applicability of ADS-IDAC to a larger range of issues.

(this page intentionally blank)

Appendix A – Diagnosis Matrix

Appendix A provides a summary table of the diagnostic matrix used for the decision-making process. Four event categories are included in the matrix: normal operating events (including control system failures), anticipated operational occurrences, design basis accidents, and functional imbalances. Ten symptom categories are provided – these are parameters that would either normally be periodically monitored by the operators or provide a high degree of diagnosticity for abnormal events. The matrix shows a summary mapping of each event to the related symptoms. For the purposes of this project, three relationship categories were used: (1) primary symptoms which are strongly associated with the event of interest and have the highest relationship value; (2) secondary symptoms which are typically associated with feedback mechanisms caused by primary symptoms and have a mid-range relationship value; and (3) tertiary symptoms which are considered less likely to appear or may be masked by the operation of automatic control systems.

Initiating Event	Event Category	Associated Parameters									
		Cont. Pressure	**Reactor Power**	T_{ave}	**PZR Water Level**	**RCS Pressure**	**RCS Flowrate**	**MF Flowrate**	**MS Flowrate**	**SG Water Level**	**SG Pressure**
Changing Steam Demand	Normal		T	S	T	T		T	P		
Controller Failure - PZR Water Level	Normal				P	T					
Controller Failure - RCS Pressure	Normal					P					
Controller Failure - RCS Temperature	Normal		T	P	S	T					
Controller Failure - SG Water Level	Normal							P		S	
Normal Steady-State Operation	Normal			P		P					
Leak – MF System	AOO							S		P	
Leak – MS System	AOO							T	P		
Leak – RCS	AOO				P	S					
Load Rejection	AOO		T	S	T	T		S	P		S
Loss of Feedwater Flow	AOO			T				P		P	
Loss of RCS Flow	AOO						P				
MS Isolation Valve Closure	AOO		T	S	T	T		S	P		S
RCS Overfill	AOO				P	S					
Reactor Trip	AOO		P	P	S	T		T	T		
ATWS - Loss of Load	Accident		T	S	T	T			P		S
Loss of Coolant Accident	Accident	P			P	P					
MF System Line Break	Accident	P		S				P		P	
MS System Line Break	Accident	P	T	S	T	T		T	P		P
Steam Generator Tube Rupture	Accident				P	S		T		P	
Energy Imbalance - PZR	Imbalance					P					
Energy Imbalance - RCS	Imbalance			P							
Energy Imbalance - SG	Imbalance										P
Mass Imbalance - RCS	Imbalance				P						
Mass Imbalance - SG	Imbalance									P	

P: Primary effects; S: Secondary effects; T: Tertiary effects.

Appendix B – ADS-IDAC Control Panel

Appendix B provides a comprehensive listing of all indicators, controls, and alarms on the ADS-IDAC control panel for a three-loop pressurized water reactor. The control panel is arranged into the following categories: parameter indicators, component states (associated with logical flags in the RELAP model), controller input devices (including simple binary state controllers and variable controllers), and alarms (associated with parameter values and component states). The specific format used for this input is described in Appendix K, the ADS-IDAC input manual.

Control_Panel_Parameter 140

```
Number_of_Data_Points_for_Rate_Calculation      120

Time                    CV_002 Value                20.0 0.0
Watchdog_Timer_1        CV_015 Value                20.0 0.0
Watchdog_Timer_2        CV_016 Value                20.0 0.0
Core_Power              CV_100 Value                20.0 0.0
SUR                     CV_491 Value                20.0 1.0
RATE_Core_Power         CV_100 Value                20.0 0.0
del_k                   CV_490 Value                20.0 0.0
Loop_A_Tcold            HV_216 Liquid_Temperature   20.0 0.0
Loop_B_Tcold            HV_316 Liquid_Temperature   20.0 0.0
Loop_C_Tcold            HV_416 Liquid_Temperature   20.0 0.0
Loop_A_Tave             CV_101 Value                20.0 0.0
Loop_B_Tave             CV_102 Value                20.0 0.0
Loop_C_Tave             CV_103 Value                20.0 0.0
Loop_A_Thot             HV_204 Liquid_Temperature   20.0 0.0
Loop_B_Thot             HV_304 Liquid_Temperature   20.0 0.0
Loop_C_Thot             HV_405 Liquid_Temperature   20.0 0.0
Loop_A_Delta_T          CV_121 Value                20.0 0.0
Loop_B_Delta_T          CV_131 Value                20.0 0.0
Loop_C_Delta_T          CV_141 Value                20.0 0.0
Tave-Tref               CV_480 Value                20.0 0.0
PZR_Pressure            HV_340 Pressure             20.0 0.0
RATE_PZR_Pressure       HV_340 Pressure             20.0 0.0
Loop_A_Pressure         HV_218 Pressure             20.0 0.0
Loop_B_Pressure         HV_318 Pressure             20.0 0.0
Loop_C_Pressure         HV_418 Pressure             20.0 0.0
Min_Sub_Cooling         CV_303 Value                20.0 0.0
HTMode_Max              CV_977 Value                20.0 0.0
Tclad_Max               CV_978 Value                20.0 0.0
PZR_Level               CV_202 Value                20.0 0.0
RATE_PZR_Level          CV_202 Value                20.0 0.0
Rx_Vessel_Level         CV_395 Value                20.0 0.0
Makeup_Flow             HJ_972 Mass_Flow_Rate       20.0 0.0
ECCS_Flow               CV_984 Value                20.0 0.0
LPI_Loop_A              HJ_943 Mass_Flow_Rate       20.0 0.0
LPI_Loop_B              HJ_944 Mass_Flow_Rate       20.0 0.0
LPI_Loop_C              HJ_945 Mass_Flow_Rate       20.0 0.0
LPI_HDR_Pressure        HV_933 Pressure             20.0 0.0
HPI_Loop_A              HJ_963 Mass_Flow_Rate       20.0 0.0
HPI_Loop_B              HJ_964 Mass_Flow_Rate       20.0 0.0
HPI_Loop_C              HJ_965 Mass_Flow_Rate       20.0 0.0
HPI_HDR_Pressure        HV_953 Pressure             20.0 0.0
ACC_A_Level             CV_390 Value                20.0 0.0
ACC_B_Level             CV_391 Value                20.0 0.0
ACC_C_Level             CV_392 Value                20.0 0.0
ACC_A_Pressure          HV_911 Pressure             20.0 0.0
ACC_B_Pressure          HV_912 Pressure             20.0 0.0
ACC_C_Pressure          HV_913 Pressure             20.0 0.0
SG_A_NR_Level           CV_506 Value                20.0 0.0
SG_B_NR_Level           CV_606 Value                20.0 0.0
SG_C_NR_Level           CV_706 Value                20.0 0.0
RATE_SG_A_NR_Level      CV_506 Value                20.0 0.0
RATE_SG_B_NR_Level      CV_606 Value                20.0 0.0
```

```
RATE_SG_C_NR_Level      CV_706 Value                 20.0 0.0
SG_A_WR_Level           CV_503 Value                 20.0 0.0
SG_B_WR_Level           CV_603 Value                 20.0 0.0
SG_C_WR_Level           CV_703 Value                 20.0 0.0
Collapsed_SG_A_WR_Level CV_535 Value                 20.0 0.0
Collapsed_SG_B_WR_Level CV_635 Value                 20.0 0.0
Collapsed_SG_C_WR_Level CV_735 Value                 20.0 0.0
SG_A_Pressure           HV_550 Pressure              20.0 0.0
SG_B_Pressure           HV_650 Pressure              20.0 0.0
SG_C_Pressure           HV_750 Pressure              20.0 0.0
RATE_SG_A_Pressure      HV_550 Pressure              20.0 0.0
RATE_SG_B_Pressure      HV_650 Pressure              20.0 0.0
RATE_SG_C_Pressure      HV_750 Pressure              20.0 0.0
Stm_HDR_Pressure        HV_802 Pressure              20.0 0.0
RATE_Stm_HDR_Pressure   HV_802 Pressure              20.0 0.0
SG_A_FW_Flow            HJ_527 Mass_Flow_Rate        20.0 0.0
SG_B_FW_Flow            HJ_627 Mass_Flow_Rate        20.0 0.0
SG_C_FW_Flow            HJ_727 Mass_Flow_Rate        20.0 0.0
RATE_SG_A_FW_Flow       HJ_527 Mass_Flow_Rate        20.0 0.0
RATE_SG_B_FW_Flow       HJ_627 Mass_Flow_Rate        20.0 0.0
RATE_SG_C_FW_Flow       HJ_727 Mass_Flow_Rate        20.0 0.0
SG_A_MS_Flow            HJ_282 Mass_Flow_Rate        20.0 0.0
SG_B_MS_Flow            HJ_382 Mass_Flow_Rate        20.0 0.0
SG_C_MS_Flow            HJ_482 Mass_Flow_Rate        20.0 0.0
RATE_SG_A_MS_Flow       HJ_282 Mass_Flow_Rate        20.0 0.0
RATE_SG_B_MS_Flow       HJ_382 Mass_Flow_Rate        20.0 0.0
RATE_SG_C_MS_Flow       HJ_482 Mass_Flow_Rate        20.0 0.0
Turb_MS_Flow            HJ_804 Mass_Flow_Rate        20.0 0.0
Turb_Gov_Vlv_Pos        CV_924 Value                 20.0 0.0
Stm_Power               CV_477 Value                 20.0 0.0
Stm_Dump_Flow           HJ_808 Mass_Flow_Rate        20.0 0.0
Stm_Dump_VPI            CV_895 Value                 20.0 0.0
CST_Level               CV_519 Value                 20.0 0.0
Turb_Pressure           CV_939 Value                 20.0 0.0
RATE_Loop_A_Tave        CV_101 Value                 20.0 0.0
RATE_Loop_B_Tave        CV_102 Value                 20.0 0.0
RATE_Loop_C_Tave        CV_103 Value                 20.0 0.0
SG_Level_Setpoint       CV_510 Value                 20.0 0.0
PZR_Level_Setpoint      CV_188 Value                 20.0 0.0
SG_A_PORV_Setpoint      CV_842 Value                 20.0 0.0
SG_B_PORV_Setpoint      CV_843 Value                 20.0 0.0
SG_C_PORV_Setpoint      CV_844 Value                 20.0 0.0
PZR_PORV_VPI            CV_589 Value                 20.0 0.0
PZR_Spray_Vlv_VPI       CV_590 Value                 20.0 0.0
SG_A_TDAFW_VPI          CV_371 Value                 20.0 0.0
SG_A_MDAFW_VPI          CV_372 Value                 20.0 0.0
SG_B_TDAFW_VPI          CV_373 Value                 20.0 0.0
SG_B_MDAFW_VPI          CV_374 Value                 20.0 0.0
SG_C_TDAFW_VPI          CV_375 Value                 20.0 0.0
SG_C_MDAFW_VPI          CV_376 Value                 20.0 0.0
SG_A_FWRV_VPI           CV_586 Value                 20.0 0.0
SG_B_FWRV_VPI           CV_587 Value                 20.0 0.0
SG_C_FWRV_VPI           CV_588 Value                 20.0 0.0
SG_A_MFIV_VPI           CV_580 Value                 20.0 0.0
SG_B_MFIV_VPI           CV_581 Value                 20.0 0.0
SG_C_MFIV_VPI           CV_582 Value                 20.0 0.0
SG_A_MSIV_VPI           CV_583 Value                 20.0 0.0
```

```
SG_B_MSIV_VPI           CV_584 Value                  20.0 0.0
SG_C_MSIV_VPI           CV_585 Value                  20.0 0.0
SG_A_Stm_Press_Rate     CV_563 Value                  20.0 0.0
SG_B_Stm_Press_Rate     CV_567 Value                  20.0 0.0
SG_C_Stm_Press_Rate     CV_571 Value                  20.0 0.0
Containment_Pressure    CV_995 Value                  20.0 0.0
Total_AFW_Flow          CV_305 Value                  20.0 0.0
Total_MFW_Flow          CV_306 Value                  20.0 0.0
MFW_Pump_A_Speed        CV_861 Value                  20.0 0.0
MFW_Pump_B_Speed        CV_864 Value                  20.0 0.0
MFP_Recirculation_Flow  HJ_868 Mass_Flow_Rate         20.0 0.0
Cond_Pump_Disch_Press   HV_854 Pressure               20.0 0.0
Median_Tave             CV_114 Value                  20.0 0.0
RATE_Median_Tave        CV_114 Value                  20.0 0.0
SG_A_PORV_VPI           CV_591 Value                  20.0 0.0
SG_B_PORV_VPI           CV_592 Value                  20.0 0.0
SG_C_PORV_VPI           CV_593 Value                  20.0 0.0
SG_A_Level_Deviation    CV_011 Value                  20.0 0.0
SG_B_Level_Deviation    CV_012 Value                  20.0 0.0
SG_C_Level_Deviation    CV_013 Value                  20.0 0.0
SGTR_Pressure           CV_946 Value                  20.0 0.0
SGTR_Temp_Target        CV_947 Value                  20.0 0.0
SGTR_A_BRK_Flow         HJ_209 Mass_Flow_Rate         20.0 0.0
SGTR_B_BRK_Flow         HJ_309 Mass_Flow_Rate         20.0 0.0
SGTR_C_BRK_Flow         HJ_409 Mass_Flow_Rate         20.0 0.0
Air_Ejector_Radiation   CV_682 Value                  20.0 0.0
PORV_Flow               HJ_344 Mass_Flow_Rate         20.0 0.0
Aux_Spay_Flow           HJ_331 Mass_Flow_Rate         20.0 0.0
MSLB_BRK_Flow           HJ_820 Mass_Flow_Rate         20.0 0.0
LOCA_BRK_Flow           HJ_995 Mass_Flow_Rate         20.0 0.0
SG_B_BRK_Flow           HJ_998 Mass_Flow_Rate         20.0 0.0
```

Indicator_Heat_Structure_value 0

Control_Panel_Component_State 73

```
Reactor_Trip                LT_1698 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
Safety_Injection            LT_1669 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Turbine_Trip                LT_1602 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Turbine_Runback             LT_1679 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Main_Steam_Isolation        LT_1663 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Block_Main_Steam_Isolation VT_0616 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Block_Low_Press_SI          LT_1561 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Permissive_P-11_PZR_Press   VT_0671 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Main_Feed_Pump_A_Trip       LT_1628 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
Main_Feed_Pump_B_Trip       LT_1629 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
MFP_Low_Suction_Pressure    VT_0529 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Main_Feed_Pump_Trip         LT_1630 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Condenser_Low_Vacuum        LT_1744 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Hi_PWR_Reactor_Trip         VT_0503 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
OTDT_Reactor_Trip           VT_0507 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
OPDT_Reactor_Trip           VT_0508 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Lo_RCS_Flow_Rx_Trip         VT_0509 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Lo_SG_Pressure_SI           LT_1665 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Lo_PZR_Pressure_SI          VT_0522 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Hi_Cont_Pressure            VT_0572 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
TDAFP_Auto_Start            LT_1634 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
MDAFP_Auto_Start            LT_1637 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
MF_MS_Mismatch_Rx_Trip      LT_1684 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
LoLo_SG_Level_Rx_Trip       VT_0539 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
```

```
Lo_Pressure_Rx_Trip        VT_0521 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
CVCS_Letdown_Isolation     LT_1507 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Hi_Pressure_Rx_Trip        VT_0520 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Hi_PZR_Level_Rx_Trip       VT_0519 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
HiHi_Cont_Pressure         VT_0568 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
TDAFP_On                   LT_1636 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
A_MDAFP_On                 LT_1642 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
B_MDAFP_On                 LT_1646 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
A_HPI_Pump_On              LT_1737 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
B_HPI_Pump_On              LT_1739 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
A_LPI_Pump_On              LT_1732 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
B_LPI_Pump_On              LT_1735 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
Hi_Neg_SG_Pressure_Rate    VT_0653 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
PZR_Prop_Htrs_On           VT_0465 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
PZR_Backup_Htrs_On         VT_0466 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
PZR_PORV_Open              VT_0655 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Spray_Vlv_A_Open           VT_0656 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Spray_Vlv_B_Open           VT_0657 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
Steam_Dump_Vlv_Open        VT_0658 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
SG_A_Safety_Open           VT_0659 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
SG_B_Safety_Open           VT_0660 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
SG_C_Safety_Open           VT_0661 Trip_Time 20.0 ON  GREATER_THEN_ON 1.0E-5
SG_A_PORV_Man              VT_0472 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
SG_B_PORV_Man              VT_0473 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
SG_C_PORV_Man              VT_0474 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
RCP_A_Tripped              LT_1745 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
RCP_B_Tripped              LT_1746 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
RCP_C_Tripped              LT_1747 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
Rods_Out                   VT_0642 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
Rods_In                    VT_0643 Trip_Time 20.0 OFF GREATER_THEN_ON 1.0E-5
FLAG_Scram                 VT_0901 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_Turb_Trip             VT_0902 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_Safety_Injection      VT_0903 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_MSIV_Trip             VT_0904 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_MDAFP_On              VT_0905 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_TDAFP_On              VT_0906 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_HPI_On                VT_0907 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_LPI_On                VT_0908 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_GOAL_Depressurize_SGA VT_0909 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_SGTR_SG_A             VT_0910 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_SGTR_SG_B             VT_0911 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_SGTR_SG_C             VT_0912 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_SG_A_Faulted          VT_0914 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_SG_B_Faulted          VT_0915 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_SG_C_Faulted          VT_0916 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_SG_Makeup_via_MFW     VT_0920 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_GOAL_Cool_Down_RCS    VT_0921 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_GOAL_Depressurize_RCS VT_0922 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
FLAG_GOAL_Reduce_Power     VT_0923 Trip_Time 1.0  ON  GREATER_THEN_ON 1.0E-5
```

Controls_Panel_Fine_Adjust 29

```
X_PZR_Spray_Valve            IC_804 1.0  1.0   0.0  -1.0  0.0
X_PZR_PORV                   IC_805 1.0  0.33  0.0  -1.0  0.035
X_SG_A_Atmos_PORV            IC_806 1.0  1.0   0.0  -1.0  0.0
X_SG_B_Atmos_PORV            IC_807 1.0  1.0   0.0  -1.0  0.0
X_SG_C_Atmos_PORV            IC_808 1.0  1.0   0.0  -1.0  0.0
X_SG_A_FWRV                  IC_809 1.0  1.0   0.0  -1.0  0.0
X_SG_B_FWRV                  IC_810 1.0  1.0   0.0  -1.0  0.0
X_SG_C_FWRV                  IC_811 1.0  1.0   0.0  -1.0  0.0
X_Stm_Dump                   IC_834 1.0  1.0   0.0  -1.0  0.0
X_SG_A_MDAFW_Throttle        IC_841 1.0  1.0   0.0  -1.0  0.0
X_SG_B_MDAFW_Throttle        IC_842 1.0  1.0   0.0  -1.0  0.0
```

```
X_SG_C_MDAFW_Throttle         IC_843 1.0  1.0   0.0  -1.0  0.0
X_SG_A_TDAFW_Throttle         IC_838 1.0  1.0   0.0  -1.0  0.0
X_SG_B_TDAFW_Throttle         IC_839 1.0  1.0   0.0  -1.0  0.0
X_SG_C_TDAFW_Throttle         IC_840 1.0  1.0   0.0  -1.0  0.0
X_LPI_Loop_A_Throttle         IC_844 1.0  1.0   0.0  -1.0  0.0
X_LPI_Loop_B_Throttle         IC_845 1.0  1.0   0.0  -1.0  0.0
X_LPI_Loop_C_Throttle         IC_846 1.0  1.0   0.0  -1.0  0.0
X_HPI_Loop_A_Throttle         IC_847 1.0  1.0   0.0  -1.0  0.0
X_HPI_Loop_B_Throttle         IC_848 1.0  1.0   0.0  -1.0  0.0
X_HPI_Loop_C_Throttle         IC_849 1.0  1.0   0.0  -1.0  0.0
X_SG_A_PORV_Setpoint          IC_867 1.0  1050. 0.0  1050. 500.0
X_SG_B_PORV_Setpoint          IC_868 1.0  1050. 0.0  1050. 1050.0
X_SG_C_PORV_Setpoint          IC_869 1.0  1050. 0.0  1050. 1050.0
X_Stm_Dump_Pressure_Setpoint  IC_870 1.0  1020. 0.0  1020. 1020.0
X_PZR_Level_Setpoint          IC_873 1.0  -1.0  0.22 -1.0  0.22
X_PZR_Aux_Spray_Valve         IC_875 1.0  1.0   0.0  -1.0  0.0
X_Watchdog_Timer_1            IC_913 1.0  9999. 0.0  -1.0  0.0
X_Watchdog_Timer_2            IC_924 1.0  9999. 0.0  -1.0  0.0
```

Control_Panel_Controller 67

```
X_MFP_Trip                    IC_801 1.0 1.0 -1.0 -1.0 ON
X_SCRAM                       IC_802 1.0 1.0  0.0 -1.0 OFF
X_Turb_Trip                   IC_803 1.0 1.0  0.0 -1.0 ON
X_Turb_Runback                IC_863 1.0 1.0  0.0 -1.0 ON
X_SG_A_FWIV                   IC_812 1.0 1.0  0.0 -1.0 ON
X_SG_B_FWIV                   IC_813 1.0 1.0  0.0 -1.0 ON
X_SG_C_FWIV                   IC_814 1.0 1.0  0.0 -1.0 ON
X_SG_A_MSIV                   IC_815 1.0 1.0  0.0 -1.0 ON
X_SG_B_MSIV                   IC_816 1.0 1.0  0.0 -1.0 ON
X_SG_C_MSIV                   IC_817 1.0 1.0  0.0 -1.0 ON
X_SIAS                        IC_818 1.0 1.0  0.0 -1.0 OFF
X_SI_BLK_A                    IC_819 1.0 1.0 -1.0 -1.0 ON
X_SI_BLK_B                    IC_820 1.0 1.0 -1.0 -1.0 ON
X_RCP_A                       IC_821 1.0 1.0 -1.0 -1.0 ON
X_RCP_B                       IC_822 1.0 1.0 -1.0 -1.0 ON
X_RCP_C                       IC_823 1.0 1.0 -1.0 -1.0 ON
X_LPI_Pump_A                  IC_824 1.0 1.0 -1.0 -1.0 ON
X_LPI_Pump_B                  IC_825 1.0 1.0 -1.0 -1.0 ON
X_HPI_Pump_A                  IC_826 1.0 1.0 -1.0 -1.0 ON
X_HPI_Pump_B                  IC_827 1.0 1.0 -1.0 -1.0 ON
X_MD_AFW_Pump_A               IC_828 1.0 1.0  0.0 -1.0 OFF
X_MD_AFW_Pump_B               IC_829 1.0 1.0  0.0 -1.0 OFF
X_TD_AFW_Pump                 IC_830 1.0 1.0  0.0 -1.0 OFF
X_LOOP                        IC_831 1.0 1.0 -1.0 -1.0 ON
X_MF_Pump_A                   IC_832 1.0 -1.0 1.0 -1.0 OFF
X_MF_Pump_B                   IC_833 1.0 -1.0 1.0 -1.0 OFF
X_LOCA                        IC_835 1.0 1.0 -1.0 -1.0 ON
X_SGTR_SG_A                   IC_836 1.0 1.0 -1.0 -1.0 ON
X_MSLB                        IC_837 1.0 1.0 -1.0 -1.0 ON
X_ACC_A_Outlet_Valve          IC_850 1.0 1.0  0.0 -1.0 ON
X_ACC_B_Outlet_Valve          IC_851 1.0 1.0  0.0 -1.0 ON
X_ACC_C_Outlet_Valve          IC_852 1.0 1.0  0.0 -1.0 ON
X_Increase_Turbine_Load       IC_853 1.0 1.0 -1.0 -1.0 ON
X_Decrease_Turbine_Load       IC_854 1.0 1.0 -1.0 -1.0 ON
X_Control_Rods_In             IC_855 1.0 1.0 -1.0 -1.0 ON
X_Control_Rods_Out            IC_856 1.0 1.0 -1.0 -1.0 ON
X_SG_B_MSLB                   IC_857 1.0 1.0 -1.0 -1.0 ON
```

```
X_SGTR_SG_B                    IC_858 1.0 1.0 -1.0 -1.0 ON
X_SGTR_SG_C                    IC_859 1.0 1.0 -1.0 -1.0 ON
X_Block_MSIV_Trip              IC_860 1.0 1.0 -1.0 -1.0 OFF
X_Block_Low_Press_SI           IC_874 1.0 1.0 -1.0 -1.0 OFF
X_PZR_Heaters                  IC_861 1.0 1.0  0.0 -1.0 OFF
X_SI_Reset                     IC_862 1.0 1.0  0.0 -1.0 ON
X_Letdown                      IC_864 1.0 1.0  0.0 -1.0 ON
X_Isolate_Charging             IC_865 1.0 1.0  0.0 -1.0 ON
X_Emergency_Borate             IC_866 1.0 1.0  0.0 -1.0 ON
X_Steam_Auxiliaries            IC_871 1.0 1.0  0.0  1.0 ON
X_Condensate_Pump_Trip         IC_872 1.0 1.0  0.0  1.0 ON
X_FLAG_Scram                   IC_901 1.0 1.0 -1.0 -1.0 ON
X_FLAG_Turbine_Trip            IC_902 1.0 1.0 -1.0 -1.0 ON
X_FLAG_Safety_Injection        IC_903 1.0 1.0 -1.0 -1.0 ON
X_FLAG_MSIV_Trip               IC_904 1.0 1.0 -1.0 -1.0 ON
X_FLAG_MDAFP_On                IC_905 1.0 1.0 -1.0 -1.0 ON
X_FLAG_TDAFP_On                IC_906 1.0 1.0 -1.0 -1.0 ON
X_FLAG_HPI_On                  IC_907 1.0 1.0 -1.0 -1.0 ON
X_FLAG_LPI_On                  IC_908 1.0 1.0 -1.0 -1.0 ON
X_FLAG_GOAL_Depressurize_SGA   IC_909 1.0 1.0 -1.0 -1.0 ON
X_FLAG_SGTR_A                  IC_910 1.0 1.0 -1.0 -1.0 ON
X_FLAG_SGTR_B                  IC_911 1.0 1.0 -1.0 -1.0 ON
X_FLAG_SGTR_C                  IC_912 1.0 1.0 -1.0 -1.0 ON
X_FLAG_SG_A_Faulted            IC_914 1.0 1.0 -1.0 -1.0 ON
X_FLAG_SG_B_Faulted            IC_915 1.0 1.0 -1.0 -1.0 ON
X_FLAG_SG_C_Faulted            IC_916 1.0 1.0 -1.0 -1.0 ON
X_FLAG_SG_Makeup_via_MFW       IC_920 1.0 1.0 -1.0 -1.0 ON
X_FLAG_GOAL_Cool_Down_RCS      IC_921 1.0 1.0 -1.0 -1.0 ON
X_FLAG_GOAL_Depressurize_RCS   IC_922 1.0 1.0 -1.0 -1.0 ON
X_FLAG_GOAL_Reduce_Power       IC_923 1.0 1.0 -1.0 -1.0 ON
```

Alarm_for_Parameter_State 37

```
A_SG_A_Lo_Level         SG_A_NR_Level           0.5 20.0 LT 0.25
A_SG_B_Lo_Level         SG_B_NR_Level           0.5 20.0 LT 0.25
A_SG_C_Lo_Level         SG_C_NR_Level           0.5 20.0 LT 0.25
A_SG_A_LoLo_Level       SG_A_NR_Level           0.5 20.0 LT 0.12
A_SG_B_LoLo_Level       SG_B_NR_Level           0.5 20.0 LT 0.12
A_SG_C_LoLo_Level       SG_C_NR_Level           0.5 20.0 LT 0.12
A_SG_A_Hi_Level         SG_A_NR_Level           0.5 20.0 GT 0.5
A_SG_B_Hi_Level         SG_B_NR_Level           0.5 20.0 GT 0.5
A_SG_C_Hi_Level         SG_C_NR_Level           0.5 20.0 GT 0.5
A_SG_A_HiHi_Level       SG_A_NR_Level           0.5 20.0 GT 0.75
A_SG_B_HiHi_Level       SG_B_NR_Level           0.5 20.0 GT 0.75
A_SG_C_HiHi_Level       SG_C_NR_Level           0.5 20.0 GT 0.75
A_SG_A_Lo_Pressure      SG_A_Pressure           0.5 20.0 LT 600.0
A_SG_B_Lo_Pressure      SG_B_Pressure           0.5 20.0 LT 600.0
A_SG_C_Lo_Pressure      SG_C_Pressure           0.5 20.0 LT 600.0
A_SG_A_Hi_Pressure      SG_A_Pressure           0.5 20.0 GT 980.0
A_SG_B_Hi_Pressure      SG_B_Pressure           0.5 20.0 GT 980.0
A_SG_C_Hi_Pressure      SG_C_Pressure           0.5 20.0 GT 980.0
A_PZR_Pressure_Lo_Dev   PZR_Pressure            0.5 20.0 LT 2185.0
A_PZR_Pressure_Hi_Dev   PZR_Pressure            0.5 20.0 GT 2285.0
A_PZR_Lo_Pressure       PZR_Pressure            0.5 20.0 LT 2100.0
A_PZR_Hi_Pressure       PZR_Pressure            0.5 20.0 GT 2300.0
A_PZR_Lo_Level          PZR_Level               0.5 20.0 LT 0.14
A_PZR_Hi_Level          PZR_Level               0.5 20.0 GT 0.92
A_Tave_Hi_Dev           Tave-Tref               0.5 20.0 GT 1.0
```

```
A_Tave_Lo_Dev            Tave-Tref                0.5 20.0 LT -1.0
A_ACC_A_Lo_Level         ACC_A_Level              0.5 20.0 LT 70.0
A_ACC_A_LoLo_Level       ACC_A_Level              0.5 20.0 LT 10.0
A_ACC_A_Lo_Pressure      ACC_A_Pressure           0.5 20.0 LT 600.0
A_ACC_B_Lo_Level         ACC_B_Level              0.5 20.0 LT 70.0
A_ACC_B_LoLo_Level       ACC_B_Level              0.5 20.0 LT 10.0
A_ACC_B_Lo_Pressure      ACC_B_Pressure           0.5 20.0 LT 600.0
A_ACC_C_Lo_Level         ACC_C_Level              0.5 20.0 LT 70.0
A_ACC_C_LoLo_Level       ACC_C_Level              0.5 20.0 LT 10.0
A_ACC_C_Lo_Pressure      ACC_C_Pressure           0.5 20.0 LT 600.0
A_Air_Ejector_Radiation Air_Ejector_Radiation    0.5 20.0 GT 10.0
A_ENDSEQ_Parameter       Core_Power               0.5 20.0 LT 0.001
```

Alarm_for_Component_State 34

```
A_Reactor_Trip                 Reactor_Trip                0.5 1.0 ON
A_Safety_Injection             Safety_Injection            0.5 20.0 ON
A_Turbine_Trip                 Turbine_Trip                0.5 20.0 ON
A_Turbine_Runback              Turbine_Runback             0.5 20.0 ON
A_RCP_A_Tripped                RCP_A_Tripped               0.5 20.0 ON
A_RCP_B_Tripped                RCP_B_Tripped               0.5 20.0 ON
A_RCP_C_Tripped                RCP_C_Tripped               0.5 20.0 ON
A_Main_Feedwater_Pump_Trip     Main_Feed_Pump_Trip         0.5 20.0 ON
A_Main_Feed_Pump_A_Trip        Main_Feed_Pump_A_Trip       0.5 20.0 ON
A_Main_Feed_Pump_B_Trip        Main_Feed_Pump_B_Trip       0.5 20.0 ON
A_MFP_Low_Suction_Pressure     MFP_Low_Suction_Pressure    0.5 20.0 ON
A_Condenser_Low_Vacuum         Condenser_Low_Vacuum        0.5 20.0 ON
A_Main_Steam_Isolation         Main_Steam_Isolation        0.5 20.0 ON
A_Hi_Power_Reactor_Trip        Hi_PWR_Reactor_Trip         0.5 20.0 ON
A_OTDT_Reactor_Trip            OTDT_Reactor_Trip           0.5 20.0 ON
A_OPDT_Reactor_Trip            OPDT_Reactor_Trip           0.5 20.0 ON
A_Lo_RCS_Flow_Reactor_Trip     Lo_RCS_Flow_Rx_Trip         0.5 20.0 ON
A_Lo_SG_Pressure_SI            Lo_SG_Pressure_SI           0.5 20.0 ON
A_Lo_PZR_Pressure_SI           Lo_PZR_Pressure_SI          0.5 20.0 ON
A_Block_Low_Press_SI           Block_Low_Press_SI          0.5 20.0 ON
A_Permissive_P-11_PZR_Press    Permissive_P-11_PZR_Press   0.5 20.0 ON
A_Hi_Cont_Pressure             Hi_Cont_Pressure            0.5 20.0 ON
A_Hi_PZR_Level_Reactor_Trip    Hi_PZR_Level_Rx_Trip        0.5 20.0 ON
A_Hi_Pressure_Reactor_Trip     Hi_Pressure_Rx_Trip         0.5 20.0 ON
A_Lo_Pressure_Reactor_Trip     Lo_Pressure_Rx_Trip         0.5 20.0 ON
A_LoLo_SG_Level_Reactor_Trip   LoLo_SG_Level_Rx_Trip       0.5 20.0 ON
A_MF_MS_Mismatch_Reactor_Trip MF_MS_Mismatch_Rx_Trip       0.5 20.0 ON
A_HiHi_Cont_Pressure           HiHi_Cont_Pressure          0.5 20.0 ON
A_CVCS_Letdown_Isolation       CVCS_Letdown_Isolation      0.5 20.0 ON
A_Rods_Out                     Rods_Out                    0.5 20.0 ON
A_Rods_In                      Rods_In                     0.5 20.0 ON
A_ENDSEQ_Component             FLAG_Scram                  0.5 20.0 ON
A_TDAFP_Auto_Start             TDAFP_Auto_Start            0.5 20.0 ON
A_MDAFP_Auto_Start             MDAFP_Auto_Start            0.5 20.0 ON
```

Alarm_for_Difference_Between_two_Values 16

```
A_SG_A_MF_Flow_Lo    SG_A_MS_Flow       0.5 20.0 SG_A_FW_Flow  350.0
A_SG_B_MF_Flow_Lo    SG_B_MS_Flow       0.5 20.0 SG_B_FW_Flow  350.0
A_SG_C_MF_Flow_Lo    SG_C_MS_Flow       0.5 20.0 SG_C_FW_Flow  350.0
A_SG_A_MF_Flow_Hi    SG_A_FW_Flow       0.5 20.0 SG_A_MS_Flow  350.0
A_SG_B_MF_Flow_Hi    SG_B_FW_Flow       0.5 20.0 SG_B_MS_Flow  350.0
A_SG_C_MF_Flow_Hi    SG_C_FW_Flow       0.5 20.0 SG_C_MS_Flow  350.0
A_SG_A_Level_Lo_Dev  SG_Level_Setpoint  0.5 20.0 SG_A_NR_Level 0.05
```

```
A_SG_B_Level_Lo_Dev  SG_Level_Setpoint  0.5 20.0 SG_B_NR_Level 0.05
A_SG_C_Level_Lo_Dev  SG_Level_Setpoint  0.5 20.0 SG_C_NR_Level 0.05
A_SG_A_Level_Hi_Dev  SG_A_NR_Level 0.5 20.0 SG_Level_Setpoint  0.05
A_SG_B_Level_Hi_Dev  SG_B_NR_Level 0.5 20.0 SG_Level_Setpoint  0.05
A_SG_C_Level_Hi_Dev  SG_C_NR_Level 0.5 20.0 SG_Level_Setpoint  0.05
A_PZR_Level_Lo_Dev   PZR_Level_Setpoint 0.5 20.0 PZR_Level   0.05
A_PZR_Level_Hi_Dev   PZR_Level     0.5 20.0 PZR_Level_Setpoint 0.05
A_SGTR_Pressure_OK   SGTR_Pressure 0.5 20.0 PZR_Pressure     0.0
A_SGTR_Temp_OK       SGTR_Temp_Target   0.5 20.0 Median_Tave 0.0
```

(this page intentionally blank)

Appendix C – Control Panel Functional Decomposition

Appendix C provides a listing of the system decomposition used in the current ADS-IDAC three loop pressurized water reactor model. The system decomposition associates every control panel item with the specific functions the item supports. A simple three digit code is used to identify system functions – the first digit refers to the high level function (mass, energy, or momentum flow), the second digit is a system designator, and the third digit provides trending information. The three digit code is used to relate each control panel item to the specific imbalance diagnosis referenced in the diagnosis matrix (see Appendix A). Therefore, the noun name for the functions provided in Appendix C must match a functional diagnosis name in the diagnosis matrix. The specific format used for this input is described in Appendix K, the ADS-IDAC input manual.

```
"Functional Decomposition:
   Function Type (First Digit)
      1 - Mass Imbalance
      2 - Energy Imbalance
      3 - Momentum Imbalance
   System Type (Second Digit)
      1 - Reactor Coolant System
      2 - Pressurizer
      3 - Steam Generator A
      4 - Steam Generator B
      5 - Steam Generator C
      6 - Secondary System (Turbine, Condenser)
      8 - Containment
   Imbalance Trend (Third Digit)
      1 - Decrease
      2 - Increase
   Special Codes
     900 - Not Applicable"

Number_of_Functional_Items 31
Mass_Imbalance_RCS_Low            111
Mass_Imbalance_PZR_Low            121
Mass_Imbalance_SG_A_Low           131
Mass_Imbalance_SG_B_Low           141
Mass_Imbalance_SG_C_Low           151
Mass_Imbalance_Secondary_Low      161
Mass_Imbalance_Containment_Low    181
Energy_Imbalance_RCS_Low          211
Energy_Imbalance_PZR_Low          221
Energy_Imbalance_SG_A_Low         231
Energy_Imbalance_SG_B_Low         241
Energy_Imbalance_SG_C_Low         251
Energy_Imbalance_Secondary_Low    261
Energy_Imbalance_Containment_Low  281
Momentum_Imbalance_RCS_High       311
Mass_Imbalance_RCS_High           112
Mass_Imbalance_PZR_High           122
Mass_Imbalance_SG_A_High          132
Mass_Imbalance_SG_B_High          142
Mass_Imbalance_SG_C_High          152
Mass_Imbalance_Secondary_High     162
Mass_Imbalance_Containment_High   182
Energy_Imbalance_RCS_High         212
Energy_Imbalance_PZR_High         222
Energy_Imbalance_SG_A_High        232
Energy_Imbalance_SG_B_High        242
Energy_Imbalance_SG_C_High        252
Energy_Imbalance_Secondary_High   262
Energy_Imbalance_Containment_High 282
Momentum_Imbalance_RCS_High       312
Not_Applicable                    900

Control_Panel_Decomposition
Time                        900    999
Watchdog_Timer_1            900    999
Watchdog_Timer_2            900    999
PZR_Level                   111    112    121    122    999
PZR_Level_Setpoint          111    112    121    122    999
RATE_PZR_Level                     111    112    121    122    999
X_PZR_Level_Setpoint               111    112    121    122    999
Rx_Vessel_Level             111    212    999
Makeup_Flow                 111    112    121    122    999
ECCS_Flow                   111    112    121    122    211    212    999
```

LPI_Loop_A		111	112	121	122	211	212	999	
LPI_Loop_B		111	112	121	122	211	212	999	
LPI_Loop_C		111	112	121	122	211	212	999	
LPI_HDR_Pressure		111	112	121	122	211	212	999	
HPI_Loop_A		111	112	121	122	211	212	999	
HPI_Loop_B		111	112	121	122	211	212	999	
HPI_Loop_C		111	112	121	122	211	212	999	
HPI_HDR_Pressure		111	112	121	122	211	212	999	
ACC_A_Level		111	112	121	122	211	212	999	
ACC_B_Level		111	112	121	122	211	212	999	
ACC_C_Level		111	112	121	122	211	212	999	
ACC_A_Pressure			111	112	121	122	211	212	999
ACC_B_Pressure			111	112	121	122	211	212	999
ACC_C_Pressure			111	112	121	122	211	212	999
PZR_Pressure		121	122	211	212	221	222	999	
PORV_Flow			121	122	221	222	999		
Aux_Spay_Flow			121	122	221	222	999		
RATE_PZR_Pressure		121	122	211	212	221	222	999	
Loop_A_Pressure		211	212	221	222	999			
Loop_B_Pressure		211	212	221	222	999			
Loop_C_Pressure		211	212	221	222	999			
RCP_A_Tripped		221	222	311	312	999			
RCP_B_Tripped			221	222	311	312	999		
RCP_C_Tripped		221	222	311	312	999			
SG_A_NR_Level		131	132	999					
SG_B_NR_Level		141	142	999					
SG_C_NR_Level		151	152	999					
SG_Level_Setpoint		131	132	141	142	151	152	999	
RATE_SG_A_NR_Level		131	132	999					
RATE_SG_B_NR_Level		141	142	999					
RATE_SG_C_NR_Level		151	152	999					
SG_A_WR_Level		131	132	999					
SG_B_WR_Level		141	142	999					
SG_C_WR_Level		151	152	999					
Collapsed_SG_A_WR_Level			131	132	999				
Collapsed_SG_B_WR_Level			141	142	999				
Collapsed_SG_C_WR_Level			151	152	999				
SG_A_Level_Deviation			131	132	999				
SG_B_Level_Deviation			141	142	999				
SG_C_Level_Deviation			151	152	999				
SG_A_Pressure		231	232	999					
SG_B_Pressure		241	242	999					
SG_C_Pressure		251	252	999					
SG_A_PORV_Setpoint			231	232	999				
SG_B_PORV_Setpoint			241	242	999				
SG_C_PORV_Setpoint			251	252	999				
Hi_Neg_SG_Pressure_Rate			231	241	251	261	999		
SG_A_Stm_Press_Rate		231	232	261	262	999			
SG_B_Stm_Press_Rate		241	242	261	262	999			
SG_C_Stm_Press_Rate		251	252	261	262	999			
RATE_SG_A_Pressure			231	232	261	262	999		
RATE_SG_B_Pressure		241	242	261	262	999			
RATE_SG_C_Pressure			251	252	261	262	999		
Stm_HDR_Pressure			261	262	999				
RATE_Stm_HDR_Pressure		261	262	999					
SG_A_FW_Flow		131	132	999					
SG_B_FW_Flow		141	142	999					
SG_C_FW_Flow		151	152	999					
Total_AFW_Flow	131	132	141	142	151	152	211	212	999
Total_MFW_Flow	131	132	141	142	151	152	211	212	999
MFW_Pump_A_Speed		131	132	141	142	151	152	999	
MFW_Pump_B_Speed		131	132	141	142	151	152	999	
MFP_Recirculation_Flow			131	132	141	142	151	152	999

Cond_Pump_Disch_Press		131	132	141	142	151	152	999
RATE_SG_A_FW_Flow	131	132	999					
RATE_SG_B_FW_Flow	141	142	999					
RATE_SG_C_FW_Flow	151	152	999					
SG_A_MS_Flow	131	132	231	232	999			
SG_B_MS_Flow	141	142	241	242	999			
SG_C_MS_Flow	151	152	251	252	999			
RATE_SG_A_MS_Flow	131	132	231	232	999			
RATE_SG_B_MS_Flow	141	142	241	242	999			
RATE_SG_C_MS_Flow	151	152	251	252	999			
Turb_MS_Flow	211	212	261	262	999			
Turb_Gov_Vlv_Pos	211	212	261	262	999			
Stm_Power	211	212	261	262	999			
Stm_Dump_Flow	211	212	261	262	999			
Stm_Dump_VPI		211	212	261	262	999		
Stm_Dump_Pressure_Mode		211	212	261	262	999		
X_SG_A_PORV_Setpoint		131	132	231	232	999		
X_SG_B_PORV_Setpoint		141	142	241	242	999		
X_SG_C_PORV_Setpoint		151	152	251	252	999		
X_Stm_Dump_Pressure_Setpoint		211	212	261	262	999		
Condenser_Low_Vacuum		211	212	261	262	999		
A_Condenser_Low_Vacuum		211	212	261	262	999		
CST_Level	161	162	999					
Turb_Pressure	261	262	999					
Core_Power	211	212	999					
SUR	211	212	999					
RATE_Core_Power	211	212	999					
del_k	211	212	999					
Loop_A_Tcold	211	212	999					
Loop_B_Tcold	211	212	999					
Loop_C_Tcold	211	212	999					
Loop_A_Thot	211	212	999					
Loop_B_Thot	211	212	999					
Loop_C_Thot	211	212	999					
Loop_A_Tave	211	212	999					
Loop_B_Tave	211	212	999					
Loop_C_Tave	211	212	999					
Loop_A_Delta_T	211	212	311	312	999			
Loop_B_Delta_T	211	212	311	312	999			
Loop_C_Delta_T	211	212	311	312	999			
Median_Tave	211	212	999					
RATE_Median_Tave	211	212	999					
RATE_Loop_A_Tave	211	212	999					
RATE_Loop_B_Tave	211	212	999					
RATE_Loop_C_Tave	211	212	999					
Tave-Tref	211	212	261	262	999			
Min_Sub_Cooling	211	212	221	222	999			
HTMode_Max	211	212	221	222	311	312	999	
Tclad_Max	211	212	221	222	311	312	999	
PZR_PORV_VPI	121	122	221	222	999			
PZR_Spray_Vlv_VPI	121	122	221	222	999			
SG_A_TDAFW_VPI		131	132	999				
SG_B_TDAFW_VPI		141	142	999				
SG_C_TDAFW_VPI		151	152	999				
SG_A_MDAFW_VPI		131	132	999				
SG_B_MDAFW_VPI		141	142	999				
SG_C_MDAFW_VPI		151	152	999				
SG_A_FWRV_VPI		131	132	999				
SG_B_FWRV_VPI		141	142	999				
SG_C_FWRV_VPI		151	152	999				
SG_A_MFIV_VPI		131	132	999				
SG_B_MFIV_VPI		141	142	999				
SG_C_MFIV_VPI		151	152	999				

SG_A_MSIV_VPI			131	132	231	232	999		
SG_B_MSIV_VPI			141	142	241	242	999		
SG_C_MSIV_VPI			151	152	251	252	999		
SG_A_PORV_VPI		131	132	231	232	999			
SG_B_PORV_VPI		141	142	241	242	999			
SG_C_PORV_VPI			151	152	251	252	999		
PZR_Prop_Htrs_On		221	222	999					
PZR_Backup_Htrs_On		221	222	999					
PZR_PORV_Open			121	122	221	222	999		
Spray_Vlv_A_Open		221	222	999					
Spray_Vlv_B_Open		221	222	999					
SG_A_Safety_Open		131	132	231	232	999			
SG_B_Safety_Open		141	142	241	242	999			
SG_C_Safety_Open		151	152	251	252	999			
SG_A_PORV_Man			131	132	231	232	999		
SG_B_PORV_Man			141	142	241	242	999		
SG_C_PORV_Man		151	152	251	252	999			
SGTR_Pressure	221	221	231	232	241	242	251	252	999
SGTR_Temp_Target	211	212	231	232	241	242	251	252	999
SGTR_A_BRK_Flow			111	112	131	132	999		
SGTR_B_BRK_Flow			111	112	141	142	999		
SGTR_C_BRK_Flow			111	112	151	152	999		
Air_Ejector_Radiation		111	121	132	142	152	999		
MSLB_BRK_Flow		131	141	151	161	261	211	212	999
LOCA_BRK_Flow			111	112	121	122	182	282	999
SG_B_BRK_Flow		141	241	211	212	182	282	999	
Steam_Dump_Vlv_Open		211	212	261	262	999			
Containment_Pressure			181	182	281	282	999		
X_RCP_A		311	312	999					
X_RCP_B			311	312	999				
X_RCP_C		311	312	999					
X_MFP_Trip		131	132	141	142	151	152	999	
X_MF_Pump_A		131	132	141	142	151	152	999	
X_MF_Pump_B			131	132	141	142	151	152	999
X_Condensate_Pump_Trip			131	132	141	142	151	152	999
X_MD_AFW_Pump_A			131	132	141	142	151	152	999
X_MD_AFW_Pump_B			131	132	141	142	151	152	999
X_TD_AFW_Pump		131	132	141	142	151	152	999	
X_SCRAM			211	212	999				
X_Control_Rods_In		211	212	999					
X_Control_Rods_Out		211	212	999					
X_LOOP			211	212	261	262	999		
X_SIAS	111	112	121	122	182	211	212	282	999
X_SI_BLK_A	111	112	121	122	182	211	212	282	999
X_SI_BLK_B	111	112	121	122	182	211	212	282	999
X_SI_Reset	111	112	121	122	182	211	212	282	999
X_Letdown		111	112	121	122	999			
X_Isolate_Charging		111	112	121	122	999			
X_Emergency_Borate		211	212	999					
X_LOCA		111	121	999					
X_SGTR_SG_A		111	121	132	999				
X_SGTR_SG_B		111	121	142	999				
X_SGTR_SG_C		111	121	152	999				
X_MSLB		211	212	261	262	999			
X_Turb_Trip		211	212	261	262	999			
X_Turb_Runback		211	212	261	262	999			
X_Stm_Dump		211	212	261	262	999			
X_PZR_Spray_Valve		221	222	999					
X_PZR_Aux_Spray_Valve			221	222	999				
X_PZR_PORV		111	112	121	122	221	222	999	
X_PZR_Heaters		221	222	999					
X_LPI_Pump_A		111	112	121	122	211	212	999	
X_LPI_Pump_B		111	112	121	122	211	212	999	

```
X_LPI_Loop_A_Throttle              111    112    121    122    211    212    999
X_LPI_Loop_B_Throttle              111    112    121    122    211    212    999
X_LPI_Loop_C_Throttle              111    112    121    122    211    212    999
X_HPI_Pump_A                       111    112    121    122    211    212    999
X_HPI_Pump_B                111    112    121    122    211    212    999
X_HPI_Loop_A_Throttle       111    112    121    122    211    212    999
X_HPI_Loop_B_Throttle       111    112    121    122    211    212    999
X_HPI_Loop_C_Throttle              111    112    121    122    211    212    999
X_ACC_A_Outlet_Valve               111    112    121    122    211    212    999
X_ACC_B_Outlet_Valve               111    112    121    122    211    212    999
X_ACC_C_Outlet_Valve               111    112    121    122    211    212    999
X_SG_A_Atmos_PORV                  131    132    231    232     999
X_SG_B_Atmos_PORV                  141    142    241    242    999
X_SG_C_Atmos_PORV                  151    152    251    252    999
X_SG_A_FWRV                 131    132     999
X_SG_B_FWRV                 141    142     999
X_SG_C_FWRV                 151    152     999
X_SG_A_TDAFW_Throttle              131    132    999
X_SG_B_TDAFW_Throttle              141    142    999
X_SG_C_TDAFW_Throttle              151    152    999
X_SG_A_MDAFW_Throttle              131    132    999
X_SG_B_MDAFW_Throttle              141    142    999
X_SG_C_MDAFW_Throttle              151    152    999
X_SG_A_FWIV                 131    132    999
X_SG_B_FWIV                 141    142    999
X_SG_C_FWIV                 151    152    999
X_SG_A_MSIV                 131    132    231    232    999
X_SG_B_MSIV                 141    142    241    242    999
X_SG_C_MSIV                 151    152    251    252    999
X_Block_MSIV_Trip           231    241    251    261    999
X_Increase_Turbine_Load            211    212    261    262    999
X_Decrease_Turbine_Load            211    212    261    262    999
X_Steam_Auxiliaries         211    212    261    262    999
X_SG_B_MSLB                 231    241    251    261    182    282    999
X_FLAG_Scram                       211    212    999
X_FLAG_Turbine_Trip         261    262    999
X_FLAG_Safety_Injection     111    112    121    122    211    212    999
X_FLAG_MSIV_Trip     231    232    241    242    251    252    261    262    999
X_FLAG_MDAFP_On      131    132    141    142    151    152    211    212    999
X_FLAG_TDAFP_On      131    132    141    142    151    152    211    212    999
X_FLAG_HPI_On                      111    112    121    122    211    212    999
X_FLAG_LPI_On                      111    112    121    122    211    212    999
X_FLAG_GOAL_Depressurize_SGA              131    132    231    232     999
X_FLAG_SGTR_A               111    121    132    999
X_FLAG_SGTR_B               111    121    142    999
X_FLAG_SGTR_C               111    121    152    999
X_FLAG_SG_A_Faulted         131    231    999
X_FLAG_SG_B_Faulted         141    241    999
X_FLAG_SG_C_Faulted         151    251    999
X_FLAG_SG_Makeup_via_MFW          131     132    141    142    151    152    999
X_FLAG_GOAL_Cool_Down_RCS          900    999
X_FLAG_GOAL_Depressurize_RCS       900    999
X_FLAG_GOAL_Reduce_Power           900    999
X_Watchdog_Timer_1          900    999
X_Watchdog_Timer_2          900    999
FLAG_Scram                  211    212    999
FLAG_Turb_Trip              261    262    999
FLAG_Safety_Injection              111    112    121    122    211    212    999
FLAG_MSIV_Trip       231    241    251    261    999
FLAG_MDAFP_On        131    132    141    142    151    152    211    212    999
FLAG_TDAFP_On        131    132    141    142    151    152    211    212    999
FLAG_HPI_On                 111    112    121    122    211    212    999
FLAG_LPI_On                 111    112    121    122    211    212    999
```

FLAG_GOAL_Depressurize_SGA				131	132	231	232	999	
FLAG_SGTR_SG_A			111	121	132	999			
FLAG_SGTR_SG_B		111	121	132	999				
FLAG_SGTR_SG_C			111	121	132	999			
FLAG_SG_A_Faulted		131	231	999					
FLAG_SG_B_Faulted		141	241	999					
FLAG_SG_C_Faulted		151	251	999					
FLAG_SG_Makeup_via_MFW			131	132	141	142	151	152	999
FLAG_GOAL_Cool_Down_RCS			900	999					
FLAG_GOAL_Depressurize_RCS			900	999					
FLAG_GOAL_Reduce_Power			900	999					
Safety_Injection			111	112	121	122	211	212	999
Reactor_Trip		211	212	999					
Rods_Out		211	212	999					
Rods_In		211	212	999					
Turbine_Trip		211	212	261	262	999			
Turbine_Runback		211	212	261	262	999			
Main_Steam_Isolation			231	241	251	261	999		
Block_Main_Steam_Isolation			231	241	251	261	999		
Main_Feed_Pump_A_Trip		131	132	141	142	151	152	999	
Main_Feed_Pump_B_Trip		131	132	141	142	151	152	999	
Main_Feed_Pump_Trip		131	132	141	142	151	152	999	
MFP_Low_Suction_Pressure		131	132	141	142	151	152	999	
Hi_PWR_Reactor_Trip			211	212	999				
OTDT_Reactor_Trip			211	212	221	999			
OPDT_Reactor_Trip			211	212	221	999			
Lo_RCS_Flow_Rx_Trip		311	999						
Lo_SG_Pressure_SI			231	241	251	261	999		
Lo_PZR_Pressure_SI			111	112	121	122	211	212	999
Block_Low_Press_SI			111	112	121	122	211	212	999
X_Block_Low_Press_SI		111	112	121	122	211	212	999	
Block_Low_Press_SI		111	112	121	122	211	212	999	
Permissive_P-11_PZR_Press			111	112	121	122	211	212	999
A_Block_Low_Press_SI			111	112	121	122	211	212	999
A_Permissive_P-11_PZR_Press			111	112	121	122	211	212	999
Hi_Cont_Pressure		182	282	999					
HiHi_Cont_Pressure		182	282	999					
CVCS_Letdown_Isolation		111	112	121	122	999			
TDAFP_Auto_Start	131	132	141	142	151	152	211	212	999
MDAFP_Auto_Start	131	132	141	142	151	152	211	212	999
MF_MS_Mismatch_Rx_Trip			131	141	151	211	212	999	
LoLo_SG_Level_Rx_Trip			131	141	151	211	212	999	
Lo_Pressure_Rx_Trip			211	212	221	999			
Hi_Pressure_Rx_Trip			211	212	222	999			
Hi_PZR_Level_Rx_Trip			122	211	212	999			
TDAFP_On	131	132	141	142	151	152	211	212	999
A_MDAFP_On	131	132	141	142	151	152	211	212	999
B_MDAFP_On	131	132	141	142	151	152	211	212	999
A_HPI_Pump_On			111	112	121	122	211	212	999
B_HPI_Pump_On			111	112	121	122	211	212	999
A_LPI_Pump_On			111	112	121	122	211	212	999
B_LPI_Pump_On			111	112	121	122	211	212	999
A_Safety_Injection		111	112	121	122	211	212	999	
A_Reactor_Trip		211	212	999					
A_Turbine_Trip		211	212	261	262	999			
A_Turbine_Runback		211	212	261	262	999			
A_RCP_A_Tripped			221	222	311	312	999		
A_RCP_B_Tripped			221	222	311	312	999		
A_RCP_C_Tripped			221	222	311	312	999		
A_TDAFP_Auto_Start	131	132	141	142	151	152	211	212	999
A_MDAFP_Auto_Start	131	132	141	142	151	152	211	212	999
A_Main_Feedwater_Pump_Trip			131	132	141	142	151	152	999
A_Main_Feed_Pump_A_Trip			131	132	141	142	151	152	999

A_Main_Feed_Pump_B_Trip		131	132	141	142	151	152	999
A_MFP_Low_Suction_Pressure	131	132	141	142	151	152	999	
A_Main_Steam_Isolation	231	241	251	261	999			
A_OTDT_Reactor_Trip	211	212	221	999				
A_OPDT_Reactor_Trip	211	212		221	999			
A_Lo_RCS_Flow_Reactor_Trip		311	999					
A_Lo_SG_Pressure_SI	231	241	251	261	999			
A_Lo_PZR_Pressure_SI	111	112	121	122	211	212	999	
A_Hi_Power_Reactor_Trip	211	212	999					
A_Hi_Cont_Pressure	182	282	999					
A_HiHi_Cont_Pressure		182	282	999				
A_CVCS_Letdown_Isolation		111	112	121	122	999		
A_MF_MS_Mismatch_Reactor_Trip		131	141	151	211	212	999	
A_LoLo_SG_Level_Reactor_Trip		131	141	151	211	212	999	
A_Lo_Pressure_Reactor_Trip		211	212	221	999			
A_Hi_Pressure_Reactor_Trip		211	212	222	999			
A_Hi_PZR_Level_Reactor_Trip		122	211	212	999			
A_Rods_Out	211	212	999					
A_Rods_In	211	212	999					
A_SG_A_Level_Lo_Dev		131	999					
A_SG_B_Level_Lo_Dev		141	999					
A_SG_C_Level_Lo_Dev		151	999					
A_SG_A_Lo_Level	131	999						
A_SG_B_Lo_Level	141	999						
A_SG_C_Lo_Level	151	999						
A_SG_A_LoLo_Level	131	999						
A_SG_B_LoLo_Level	141	999						
A_SG_C_LoLo_Level	151	999						
A_SG_A_Level_Hi_Dev		132	999					
A_SG_B_Level_Hi_Dev		142	999					
A_SG_C_Level_Hi_Dev		152	999					
A_SG_A_Hi_Level	132	999						
A_SG_B_Hi_Level	142	999						
A_SG_C_Hi_Level	152	999						
A_SG_A_HiHi_Level	132	999						
A_SG_B_HiHi_Level	142	999						
A_SG_C_HiHi_Level	152	999						
A_SG_A_MF_Flow_Lo	131	999						
A_SG_B_MF_Flow_Lo	141	999						
A_SG_C_MF_Flow_Lo	151	999						
A_SG_A_MF_Flow_Hi	132	999						
A_SG_B_MF_Flow_Hi	142	999						
A_SG_C_MF_Flow_Hi	152	999						
A_SG_A_Lo_Pressure	231	999						
A_SG_B_Lo_Pressure	241	999						
A_SG_C_Lo_Pressure	251	999						
A_SG_A_Hi_Pressure	232	999						
A_SG_B_Hi_Pressure	242	999						
A_SG_C_Hi_Pressure	252	999						
A_PZR_Lo_Pressure	211	221	999					
A_PZR_Hi_Pressure	212	222	999					
A_PZR_Pressure_Hi_Dev		212	222	999				
A_PZR_Pressure_Lo_Dev		211	221	999				
A_PZR_Lo_Level	111	121	999					
A_PZR_Level_Lo_Dev	111	121	999					
A_PZR_Hi_Level	112	122	999					
A_PZR_Level_Hi_Dev	112	122	999					
A_Tave_Hi_Dev	212	999						
A_Tave_Lo_Dev	211	999						
A_ACC_A_Lo_Level	111	121	211	212	999			
A_ACC_A_LoLo_Level		111	121	211	212	999		
A_ACC_A_Lo_Pressure		111	121	211	212	999		
A_ACC_B_Lo_Level		111	121	211	212	999		

```
A_ACC_B_LoLo_Level               111    121    211    212    999
A_ACC_B_Lo_Pressure              111    121    211    212    999
A_ACC_C_Lo_Level                 111    121    211    212    999
A_ACC_C_LoLo_Level               111    121    211    212    999
A_ACC_C_Lo_Pressure              111    121    211    212    999
A_SGTR_Pressure_OK  221    222    231    232    241    242    251    252    999
A_SGTR_Temp_OK      211    212    231    232    241    242    251    252    999
A_Air_Ejector_Radiation    111    121    132    142    152    999
A_ENDSEQ_Parameter         900    999
A_ENDSEQ_Component         900    999
A_ENDSEQ_Main_Steam_Isolation           900    999
```

(this page intentionally blank)

Appendix D – Operator Mental Beliefs

Appendix D provides a representative listing of operator mental beliefs used in the current ADS-IDAC three loop pressurized water reactor model. Mental beliefs support three primary functions in ADS-IDAC: (1) provide symptom confidence levels for use in the diagnostic module, (2) activate simple skill- or rule-based actions, and (3) provide the building blocks for more complex mental models of plant behavior. Each mental belief includes prerequisite conditions (e.g., the parameter, component, alarm, mental belief and procedure states that are required to activate the belief), activation parameters (i.e., the probability that the belief is activated when the prerequisite conditions are met), timing parameters that establish the activation time and reset time for the belief, and the skill- or rule-based actions associated with the belief. The listing in the Appendix is not complete – the ADS-IDAC model includes approximately 90 mental beliefs for both the Decision-Maker and the Action-Taker. The specific format used for this input is described in Appendix K, the ADS-IDAC input manual.

```
1 Containment_Pressure_Increase
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state            1
            A_Hi_Cont_Pressure          3022
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None

2 Power_Increase
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state            1
            A_Hi_Power_Reactor_Trip     3022
      Number_of_expected_component_state        0
      Number_of_expected_parameter_value        2
            SUR               3007  0.2   0.0
            RATE_Core_Power   3007  0.5   0.0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None

3 Power_Decrease
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state            0
      Number_of_expected_component_state        0
      Number_of_expected_parameter_value        2
            SUR               3009  -0.2  0.0
            RATE_Core_Power   3009  -0.5  0.0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None

4 Tave_Increase
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state            1
```

```
      A_Tave_Hi_Dev           3022
Number_of_expected_component_state          0
Number_of_expected_parameter_value          2
      Tave-Tref         3007  2.0   0.0
      RATE_Median_Tave  3007  1.5   0.0
Number_of_manipulative_control      0
Number_of_mental_belief             0
Number_of_procedure_activity        0
Mental_procedure_priority     5
None  None
```

5 Tave_Steady

```
activation_probability  0.0
branch_probability      0.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        3600.0      1.0   1.0
Number_of_expected_alarm_state      0
Number_of_expected_component_state  0
Number_of_expected_parameter_value  2
      Tave-Tref         3032  -0.5  0.5
      RATE_Median_Tave  3032  -0.5  0.5
Number_of_manipulative_control      0
Number_of_mental_belief             0
Number_of_procedure_activity        0
Mental_procedure_priority     5
None  None
```

6 Tave_Decrease

```
activation_probability  0.0
branch_probability      0.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        3600.0      1.0   1.0
Number_of_expected_alarm_state      1
      A_Tave_Lo_Dev           3022
Number_of_expected_component_state  0
Number_of_expected_parameter_value  2
      Tave-Tref         3009  -2.0  0.0
      RATE_Median_Tave  3009  -1.5  0.0
Number_of_manipulative_control      0
Number_of_mental_belief             0
Number_of_procedure_activity        0
Mental_procedure_priority     5
None  None
```

7 Pressurizer_Level_Increase

```
activation_probability  0.0
branch_probability      0.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        3600.0      1.0   1.0
Number_of_expected_alarm_state      2
      A_PZR_Level_Hi_Dev      3022
      A_PZR_Hi_Level          3022
Number_of_expected_component_state  0
Number_of_expected_parameter_value  1
      RATE_PZR_Level    3007  0.01  0.0
Number_of_manipulative_control      0
Number_of_mental_belief             0
```

```
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None

8 Pressurizer_Level_Decrease
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state      2
            A_PZR_Level_Lo_Dev      3022
            A_PZR_Lo_Level          3022
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  1
            RATE_PZR_Level    3009  -0.01 0.0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None

9 RCS_Pressure_Increase
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state      1
            A_PZR_Hi_Pressure 3022
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  1
            RATE_PZR_Pressure 3007  25.0  0.0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None

10 RCS_Pressure_Steady
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state            0
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  1
            RATE_PZR_Pressure 3032  -5.0  5.0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None

11 RCS_Pressure_Decrease
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
```

```
      Number_of_expected_alarm_state      1
            A_PZR_Lo_Pressure 3022
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  1
            RATE_PZR_Pressure 3009  -25.0 0.0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None
```

12 SG_A_Level_Increase

```
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state      2
            A_SG_A_Level_Hi_Dev     3022
            A_SG_A_Hi_Level         3022
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  2
            RATE_SG_A_NR_Level      3007  0.01  0.0
            SG_A_Level_Deviation    3007  2.0   0.0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None
```

13 SG_A_Level_Decrease

```
      activation_probability  0.0
      branch_probability      0.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        3600.0      1.0   1.0
      Number_of_expected_alarm_state      2
            A_SG_A_Lo_Level   3022
            A_SG_A_LoLo_Level 3022
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  2
            RATE_SG_A_NR_Level      3009  -0.1  0.0
            SG_A_Level_Deviation    3009  -1.0  0.0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      None  None
```

14 CSF_Loss_of_Secondary_Heat_Sink

```
      activation_probability  0.9
      branch_probability      1.0
      activation_delay_time   0.0         100.0 1.5
      reset_delay_time        10000.0     1.0   1.0
      Number_of_expected_alarm_state      0
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  4
            SG_A_NR_Level           3009  0.10  0.0
```

```
      SG_B_NR_Level          3009  0.10  0.0
      SG_C_NR_Level          3009  0.10  0.0
      Total_AFW_Flow         3009  55.1  0.0
Number_of_manipulative_control     0
Number_of_mental_belief            1
      Monitor_Critical_Safety_Functions   3021
Number_of_procedure_activity       0
Mental_procedure_priority    1
MPBG_FRG_H.1           Step_1
```

15 ES-0.1_Foldout_SI_Actuation

```
activation_probability  0.8
branch_probability      1.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        10000.0     1.0   1.0
Number_of_expected_alarm_state     1
      A_Safety_Injection     3023
Number_of_expected_component_state  0
Number_of_expected_parameter_value  1
      PZR_Level   3009  0.1   0.0
Number_of_manipulative_control     0
Number_of_mental_belief            0
Number_of_procedure_activity       1
      ES_0.1           3126
Mental_procedure_priority    5
MPBG_SI_Actuation Step_1
```

16 ES-0.1_Return_to_E-0

```
activation_probability  0.8
branch_probability      1.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        10000.0     1.0   1.0
Number_of_expected_alarm_state     1
      A_Safety_Injection     3022
Number_of_expected_component_state  0
Number_of_expected_parameter_value  0
Number_of_manipulative_control     0
Number_of_mental_belief            0
Number_of_procedure_activity       1
      ES_0.1           3126
Mental_procedure_priority    5
MPBG_Return_to_E_0     Step_1
```

17 Reactor_Coolant_System_Leak

```
activation_probability  0.8
branch_probability      1.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        3600.0      1.0   1.0
Number_of_expected_alarm_state     2
      A_PZR_Level_Lo_Dev     3022
      A_PZR_Pressure_Lo_Dev  3022
Number_of_expected_component_state  0
Number_of_expected_parameter_value  2
      RATE_PZR_Level    3009  -0.01 0.0
      RATE_PZR_Pressure 3009  -15.0 0.0
Number_of_manipulative_control     0
Number_of_mental_belief            0
```

```
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      MPBG_RCS_Leakage        Step_1

18 High_Secondary_Radiation
      activation_probability  0.8
      branch_probability      1.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        10000.0     1.0   1.0
      Number_of_expected_alarm_state      1
            A_Air_Ejector_Radiation       3022
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      MPBG_High_Secondary_Radiation Step_1

19 Possible_SG_Tube_Rupture
      activation_probability  0.5
      branch_probability      1.0
      activation_delay_time   0.0   1.0   1.0
      reset_delay_time        300.0 1.0   1.0
      Number_of_expected_alarm_state      0
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  0
      Number_of_manipulative_control      0
      Number_of_mental_belief             3
            Reactor_Coolant_System_Leak         3021
            High_Secondary_Radiation            3021
            SG_Uncontrolled_Level_Increase      3021
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      NONE  NONE

20 Reactor_Tripped
      activation_probability  0.8
      branch_probability      1.0
      activation_delay_time   0.0         1.0   1.0
      reset_delay_time        10000.0     1.0   1.0
      Number_of_expected_alarm_state      1
            A_Reactor_Trip    3022
      Number_of_expected_component_state  0
      Number_of_expected_parameter_value  0
      Number_of_manipulative_control      0
      Number_of_mental_belief             0
      Number_of_procedure_activity        0
      Mental_procedure_priority     5
      NONE  NONE

21 Normal_Operation
      activation_probability  0.8
      branch_probability      0.0
      activation_delay_time   0.0   1.0   1.0
      reset_delay_time        10.0  1.0   1.0
      Number_of_expected_alarm_state      0
```

```
Number_of_expected_component_state  0
Number_of_expected_parameter_value  3
      Tave-Tref         3032  -1.5        1.5
      PZR_Pressure      3032  2200.0      2300.0
      PZR_Level         3032  0.40        0.55
Number_of_manipulative_control      0
Number_of_mental_belief             0
Number_of_procedure_activity        0
Mental_procedure_priority     5
NONE  NONE
```

22 Manual_SI_Actuation_Low_PZR_Level

```
activation_probability  0.8
branch_probability      0.0
activation_delay_time   0.0   1.0   1.0
reset_delay_time  10000.0     1.0   1.0
Number_of_expected_alarm_state            1
A_Safety_Injection      3023
Number_of_expected_component_state  0
Number_of_expected_parameter_value  1
PZR_Level   3009  0.12  0.0
Number_of_manipulative_control          0
Number_of_mental_belief                 1
Reactor_Tripped         3045
Number_of_procedure_activity        0
Mental_procedure_priority     5
MPBG_SI_Actuation Step_1
```

23 Manual_SI_Actuation_RCS_Leak

```
activation_probability  0.8
branch_probability      0.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        10000.0     1.0   1.0
Number_of_expected_alarm_state      1
      A_Safety_Injection      3023
Number_of_expected_component_state  0
Number_of_expected_parameter_value  0
Number_of_manipulative_control      0
Number_of_mental_belief             1
      Reactor_Coolant_System_Leak   3021
Number_of_procedure_activity        0
Mental_procedure_priority     5
MPBG_SI_Actuation Step_1
```

24 ES-1.1_Transfer_to_E-3

```
activation_probability  0.8
branch_probability      1.0
activation_delay_time   0.0   1.0   1.0
reset_delay_time        100.0 1.0   1.0
Number_of_expected_alarm_state      0
Number_of_expected_component_state  0
Number_of_expected_parameter_value  0
Number_of_manipulative_control      0
Number_of_mental_belief             1
      SG_Uncontrolled_Level_Increase      3021
Number_of_procedure_activity        1
```

```
      ES_1.1            3126
  Mental_procedure_priority     2
  MPBG_Transfer_to_E_3    Step_1
```

25 SG_A_Uncontrolled_Level_Increase

```
  activation_probability  0.70
  branch_probability      1.0
  activation_delay_time   0.0          1.0   1.0
  reset_delay_time        5000.0       1.0   1.0
  Number_of_expected_alarm_state       1
      A_SG_A_Hi_Level         3022
  Number_of_expected_component_state  0
  Number_of_expected_parameter_value  3
      RATE_SG_A_NR_Level      3007  0.02  0.0
      SG_A_Level_Deviation    3007  2.0   0.0
      SG_A_FW_Flow            3009  15.0  0.0
  Number_of_manipulative_control      0
  Number_of_mental_belief             0
  Number_of_procedure_activity        0
  Mental_procedure_priority     5
  None  None
```

26 SG_Uncontrolled_Level_Increase

```
  activation_probability  0.2
  branch_probability      1.0
  activation_delay_time   0.0          1.0   1.0
  reset_delay_time        5000.0       1.0   1.0
  Number_of_expected_alarm_state      0
  Number_of_expected_component_state  0
  Number_of_expected_parameter_value  0
  Number_of_manipulative_control      0
  Number_of_mental_belief             3
      SG_A_Uncontrolled_Level_Increase    3021
      SG_B_Uncontrolled_Level_Increase    3021
      SG_C_Uncontrolled_Level_Increase    3021
  Number_of_procedure_activity        0
  Mental_procedure_priority     5
  None  None
```

27 RCS_Leakage_Reduce_Reactor_Power

```
  activation_probability  0.8
  branch_probability      1.0
  activation_delay_time   100.0        1.0   1.0
  reset_delay_time        10000.0      1.0   1.0
  Number_of_expected_alarm_state      0
  Number_of_expected_component_state  0
  Number_of_expected_parameter_value  0
  Number_of_manipulative_control      0
  Number_of_mental_belief             2
      Reactor_Coolant_System_Leak   3021
      Reactor_Tripped               3045
  Number_of_procedure_activity        0
  Mental_procedure_priority     1
  MPBG_Reduce_Power Step_1
```

28 Isolate_SG_A

```
    activation_probability  0.8
    branch_probability      1.0
    activation_delay_time   100.0       1.0   1.0
    reset_delay_time        10000.0     1.0   1.0
    Number_of_expected_alarm_state      0
    Number_of_expected_component_state  0
    Number_of_expected_parameter_value  0
    Number_of_manipulative_control      0
    Number_of_mental_belief             3
          SG_A_Uncontrolled_Level_Increase          3021
          Possible_SG_Tube_Rupture                  3021
          Emergency_Operating_Procedures_Initiated  3045
    Number_of_procedure_activity        0
    Mental_procedure_priority     1
    MPBG_Isolate_SG_A       Step_1
```

29 Monitor_Critical_Safety_Functions

```
    activation_probability  0.1
    branch_probability      1.0
    activation_delay_time   0.0         1.0   1.0
    reset_delay_time        10000.0     1.0   1.0
    Number_of_expected_alarm_state      0
    Number_of_expected_component_state  0
    Number_of_expected_parameter_value  0
    Number_of_manipulative_control      0
    Number_of_mental_belief             0
    Number_of_procedure_activity        7
          E_0_CSF     3126
          E_1         3126
          E_2         3126
          E_3         3126
          ES_0.1      3126
          ES_1.1      3126
          FRG_H.1     3126
    Mental_procedure_priority     2
    None        None
```

30 Queue_Tripping_All_RCPs

```
    activation_probability  0.1
    branch_probability      1.0
    activation_delay_time   0.0         1.0   2.0
    reset_delay_time        3600.0      1.0   1.0
    Number_of_expected_alarm_state      3
          A_RCP_A_High_Vibration  3022
          A_RCP_B_High_Vibration  3022
          A_RCP_C_High_Vibration  3022
    Number_of_expected_component_state  0
    Number_of_expected_parameter_value  0
    Number_of_manipulative_control      0
    Number_of_mental_belief             1
          ECCS_Injection_With_Loss_of_Subcooling_Margin   3021
    Number_of_procedure_activity        1
          FRG_H.1_Immediate_Feed  3126
    Mental_procedure_priority     1
    MPBG_Stop_RCPS          Step_1
```

31 ECCS_Injection_With_Loss_of_Subcooling_Margin

```
activation_probability  0.8
branch_probability      1.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        5000.0      1.0   1.0
Number_of_expected_alarm_state      0
Number_of_expected_component_state  0
Number_of_expected_parameter_value  2
      ECCS_Flow         3007  10.0    0.0
      Min_Sub_Cooling   3009  10.0    0.0
Number_of_manipulative_control      0
Number_of_mental_belief             0
Number_of_procedure_activity        0
Mental_procedure_priority     2
NONE  NONE
```

(this page intentionally blank)

Appendix E – Sample ADS-IDAC Scenario Timeline

Appendix E provides a sample sequence event printout for the first 500 seconds of an uncomplicated reactor trip scenario. The operators establish a goal of maintaining normal operation using the wait and monitor strategy at approximately 60 seconds. A reactor trip initiating event is actuated at approximately 180 seconds. The operators then transition to the maintaining global safety margin goal at approximately 250 seconds (the troubleshooting goal is set to be bypassed). This goal shift then activates the follow-procedure strategy and results in the operators initiating emergency operating procedure E-0 at 252 seconds. Because the trip was uncomplicated, the operators transition to supplemental procedure ES-0.1 at 281 seconds to complete the stabilization of plant conditions. Of particular note is the rich narrative detail provided by the summary report including detailed procedure step actions and expectation evaluation, activation of mental beliefs, and crew communication.

```
*********************************************
*** Event Hightlights for Sequence 0 ***
*********************************************
61.0627       GOAL: Sender: ODM; Receiver: ODM; Goal: GNOP
62.0704       STRATEGY: Sender: ODM; Receiver: ODM; Strategy: SWM
180.975       HWReliability: Sender: PLANT; Receiver: PLANT; Component Reactor_Trip; Status: Fail; X_SCRAM
181.479       2 branches are generated
181.982       HWReliability: Sender: PLANT; Receiver: PLANT; Component Reactor_Trip; Status: Success; X_Stm_Dump
181.982       New Alarm:  Actuation A_Tave_Hi_Dev
              New Alarm:  Actuation A_Reactor_Trip
              New Alarm:  Actuation A_Turbine_Trip
182.485       Mental Belief: Sender: ODM; Receiver: ODM; Belief: Reactor_Tripped; Status: SUCCEED
              Mental Belief: Sender: OAT; Receiver: OAT; Belief: Reactor_Tripped; Status: SUCCEED
182.988       Info_gather_mode: Sender: OAT; Receiver: OATconditional_use_of_memorized_info_if_available
              Mental Belief: Sender: ODM; Receiver: ODM; Belief: Alarm_Hi_Tave_Deviation; Status: SUCCEED
              New Alarm:  Actuation A_SG_A_MF_Flow_Hi
              New Alarm:  Actuation A_SG_B_MF_Flow_Hi
              New Alarm:  Actuation A_SG_C_MF_Flow_Hi
183.489       Mental Belief: Sender: ODM; Receiver: ODM; Belief: Alarm_Reactor_Trip; Status: SUCCEED
183.991       GOAL: Sender: OAT; Receiver: OAT; Goal: GNOP
              Procedure: Sender: ODM; Receiver: ODM; Procedure: MPBG_ARP_High_Tave_Deviation;  Step_1
184.492       AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: Tave-Tref
              New Alarm:  Actuation A_PZR_Pressure_Lo_Dev
185.495       AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: Median_Tave
              New Alarm:  Actuation A_PZR_Level_Hi_Dev
186.498       AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: RATE_Median_Tave
186.999       New Alarm:  Actuation A_PZR_Lo_Pressure
187.499       Mental Belief: Sender: ODM; Receiver: ODM; Belief: Alarm_Low_Pressurizer_Pressure; Status: SUCCEED
188.000       New Alarm:  Actuation A_SG_A_Level_Lo_Dev
              New Alarm:  Actuation A_SG_B_Level_Lo_Dev
188.501       Procedure: Sender: ODM; Receiver: ODM; Procedure: MPBG_ARP_PZR_Low_Pressure;  Step_1
              New Alarm:  Actuation A_SG_C_Level_Lo_Dev
189.001       AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: PZR_Pressure
190.002       AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: RATE_PZR_Pressure
191.003       New Alarm:  Actuation A_SG_A_Lo_Level
              New Alarm:  Actuation A_SG_B_Lo_Level
191.503       Mental Belief: Sender: ODM; Receiver: ODM; Belief: Alarm_SG_A_Low_Level; Status: SUCCEED
              Mental Belief: Sender: OAT; Receiver: OAT; Belief: Alarm_SG_A_Low_Level; Status: SUCCEED
              New Alarm:  Actuation A_SG_C_Lo_Level
192.003       Mental Belief: Sender: ODM; Receiver: ODM; Belief: Alarm_SG_B_Low_Level; Status: SUCCEED
```

```
               Mental Belief: Sender: OAT; Receiver: OAT; Belief: Alarm_SG_B_Low_Level; Status: SUCCEED
192.503        Mental Belief: Sender: ODM; Receiver: ODM; Belief: Alarm_SG_C_Low_Level; Status: SUCCEED
               Mental Belief: Sender: OAT; Receiver: OAT; Belief: Alarm_SG_C_Low_Level; Status: SUCCEED
               New Alarm:  Actuation A_SG_A_LoLo_Level
               New Alarm:  Actuation A_SG_B_LoLo_Level
               New Alarm:  Actuation A_LoLo_SG_Level_Reactor_Trip
               New Alarm:  Actuation A_TDAFP_Auto_Start
               New Alarm:  Actuation A_MDAFP_Auto_Start
193.032        Mental Belief: Sender: OAT; Receiver: OAT; Belief: Alarm_MDAFP_Auto_Start; Status: SUCCEED
               New Alarm:  Actuation A_SG_C_LoLo_Level
               New Alarm:  Actuation A_SG_C_Hi_Pressure
193.561        Mental Belief: Sender: OAT; Receiver: OAT; Belief: Alarm_TDAFP_Auto_Start; Status: SUCCEED
               New Alarm:  Actuation A_SG_A_Hi_Pressure
               New Alarm:  Actuation A_SG_B_Hi_Pressure
194.091        Procedure: Sender: ODM; Receiver: ODM; Procedure: MPBG_ARP_SG_A_Lo_Level;  Step_1
               Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_ARP_SG_B_Lo_Level;  Step_1
194.62         AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: SG_A_NR_Level
               AddScannedParameter: Sender: OAT; Receiver: OAT; Parameter: SG_B_NR_Level
195.12         New Alarm:  Clear A_PZR_Level_Hi_Dev
195.62         AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: SG_A_WR_Level
196.12         Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_ARP_SG_C_Lo_Level;  Step_1
196.62         AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: RATE_SG_A_NR_Level
               AddScannedParameter: Sender: OAT; Receiver: OAT; Parameter: SG_C_NR_Level
198.121        Procedure: Sender: ODM; Receiver: ODM; Procedure: MPBG_ARP_SG_B_Lo_Level;  Step_1
               Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_ARP_SG_A_Lo_Level;  Step_1
198.621        AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: SG_B_NR_Level
               AddScannedParameter: Sender: OAT; Receiver: OAT; Parameter: SG_A_NR_Level
199.622        AddScannedParameter: Sender: OAT; Receiver: OAT; Parameter: SG_A_WR_Level
               New Alarm:  Clear A_SG_C_MF_Flow_Hi
200.122        Procedure: Sender: ODM; Receiver: ODM; Procedure: MPBG_ARP_SG_C_Lo_Level;  Step_1
               New Alarm:  Clear A_SG_A_MF_Flow_Hi
               New Alarm:  Clear A_SG_B_MF_Flow_Hi
200.622        AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: SG_C_NR_Level
               AddScannedParameter: Sender: OAT; Receiver: OAT; Parameter: RATE_SG_A_NR_Level
214.624        Procedure: Sender: ODM; Receiver: ODM; Procedure: MPBG_ARP_Reactor_Trip;  Step_1
215.124        AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: Core_Power
               AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: SUR
               AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: Total_AFW_Flow
242.272        GOAL: Sender: ODM; Receiver: ODM; Goal: GTS - GOAL BYPASSED
243.301        GOAL: Sender: ODM; Receiver: ODM; Goal: GMON
```

244.36 **GOAL: Sender: OAT; Receiver: OAT; Goal: GMON**
Info_gather_mode: Sender: ODM; Receiver: ODMconditional_use_of_memorized_info_if_available
245.419 **GOAL: Sender: ODM; Receiver: ODM; Goal: GMGSM**
246.478 **GOAL: Sender: OAT; Receiver: OAT; Goal: GMGSM**
STRATEGY: Sender: ODM; Receiver: ODM; Strategy: SFP
247.537 **STRATEGY: Sender: OAT; Receiver: OAT; Strategy: SFI**
251.772 **Procedure: Sender: ODM; Receiver: ODM; Procedure: E_0; Step_1**
252.301 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Emergency_Operating_Procedures_Initiated; Status: SUCCEED
252.83 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Uncomplicated_Trip_Recovery; Status: SUCCEED
253.889 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify A_Reactor_Trip ; Verify_Reactor_Trip
254.419 Alarm: Sender: ODM; Receiver: ODM Alarm: A_Reactor_Trip; Status: SUCCEED; Exp State: ON; Observed State: ON;Use of Memorized Info
255.477 Parameter: Sender: ODM; Receiver: ODM; Parameter: SUR; Status: SUCCEED; Exp State: <= -0.3; Observed value: -0.6140919539; Use of memorized info
257.595 Procedure: Sender: ODM; Receiver: ODM; Procedure: E_0; Step_2
258.654 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify A_Turbine_Trip ; Verify_Turbine_Trip
259.183 Alarm: Sender: ODM; Receiver: ODM Alarm: A_Turbine_Trip; Status: SUCCEED; Exp State: ON; Observed State: ON;Use of Memorized Info
261.301 Procedure: Sender: ODM; Receiver: ODM; Procedure: E_0; Step_3
261.83 Procedure: Sender: ODM; Receiver: ODM; Procedure: E_0; Step_4.1
262.889 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify A_Safety_Injection ; Check_Safety_Injection_Status
263.947 Alarm: Sender: ODM; Receiver: OAT Alarm: A_Lo_SG_Pressure_SI; Status: ACTIVE; Exp State: OFF; Observed State: NONE
265.006 Alarm: Sender: OAT; Receiver: PLANT Alarm: A_Lo_SG_Pressure_SI; Status: ACTIVE; Exp State: OFF; Observed State: NONE
266.065 Alarm: Sender: OAT; Receiver: ODM Alarm: A_Lo_SG_Pressure_SI; Status: ACTIVE; Exp State: OFF; Observed State: OFF
266.594 Alarm: Sender: ODM; Receiver: ODM Alarm: A_Lo_SG_Pressure_SI; Status: SUCCEED; Exp State: OFF; Observed State: OFF
268.182 Alarm: Sender: ODM; Receiver: OAT Alarm: A_Hi_Cont_Pressure; Status: ACTIVE; Exp State: OFF; Observed State: NONE
269.241 Alarm: Sender: OAT; Receiver: PLANT Alarm: A_Hi_Cont_Pressure; Status: ACTIVE; Exp State: OFF; Observed State: NONE
270.3 Alarm: Sender: OAT; Receiver: ODM Alarm: A_Hi_Cont_Pressure; Status: ACTIVE; Exp State: OFF; Observed State: OFF
270.829 Alarm: Sender: ODM; Receiver: ODM Alarm: A_Hi_Cont_Pressure; Status: SUCCEED; Exp State: OFF; Observed State: OFF

272.417 Alarm: Sender: ODM; Receiver: OAT Alarm: A_Lo_PZR_Pressure_SI; Status: ACTIVE; Exp State: OFF; Observed State: NONE
273.476 Alarm: Sender: OAT; Receiver: PLANT Alarm: A_Lo_PZR_Pressure_SI; Status: ACTIVE; Exp State: OFF; Observed State: NONE
274.535 Alarm: Sender: OAT; Receiver: ODM Alarm: A_Lo_PZR_Pressure_SI; Status: ACTIVE; Exp State: OFF; Observed State: OFF
275.064 Alarm: Sender: ODM; Receiver: ODM Alarm: A_Lo_PZR_Pressure_SI; Status: SUCCEED; Exp State: OFF; Observed State: OFF
276.652 Component: Sender: ODM; Receiver: OAT; Component: Safety_Injection; Status: ACTIVE; Exp State: OFF; Observed State: NONE
277.711 Component: Sender: OAT; Receiver: PLANT; Component: Safety_Injection; Status: ACTIVE; Exp State: OFF; Observed State: NONE
278.77 Component: Sender: OAT; Receiver: ODM; Component: Safety_Injection; Status: ACTIVE; Exp State: OFF; Observed State: OFF
279.299 Component: Sender: ODM; Receiver: ODM; Component: Safety_Injection; Status: SUCCEED; Exp State: OFF; Observed State: OFF
286.181 Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_1.1
286.71 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Monitor_Critical_Safety_Functions; Status: SUCCEED
287.239 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Uncomplicated_Trip_Recovery; Status: SUCCEED
291.474 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify Median_Tave ; Check_Reactor_Coolant_System_Temperature Steady_or_trending_to_no-load_Tave
292.003 Parameter: Sender: ODM; Receiver: ODM; Parameter: Median_Tave; Status: FAILED; Exp State: BETWEEN 546. and 548.; Observed value: 550.421549344; Use of memorized info
293.591 Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_ARP_MDAFP_Start; Step_1
294.121 ACTION: Sender: OAT; Receiver: OAT; Action: WAITING: Verify A_MDAFP_On ; Motor_Driven_Aux_Feed_Pump_Auto_Start
294.65 Component: Sender: OAT; Receiver: PLANT; Component: A_MDAFP_On; Status: ACTIVE; Exp State: OFF; Observed State: NONE
295.179 Mental Belief: Sender: OAT; Receiver: OAT; Belief: Align_A_MDAFP_Flow_Path; Status: SUCCEED
295.709 Component: Sender: OAT; Receiver: OAT; Component: A_MDAFP_On; Status: FAILED; Exp State: OFF; Observed State: ON
297.297 Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_ARP_MDAFP_Start; Step_2
297.826 ACTION: Sender: OAT; Receiver: OAT; Action: WAITING: Verify B_MDAFP_On ; Motor_Driven_Aux_Feed_Pump_Auto_Start
298.356 Component: Sender: OAT; Receiver: PLANT; Component: B_MDAFP_On; Status: ACTIVE; Exp State: OFF; Observed State: NONE
298.885 Component: Sender: OAT; Receiver: OAT; Component: B_MDAFP_On; Status: FAILED; Exp State: OFF; Observed State: ON
305.237 Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_NONE; Step_1
305.766 AddScannedParameter: Sender: OAT; Receiver: OAT; Parameter: PZR_Level

306.825	Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_ARP_TDAFP_Start; Step_1
307.354	ACTION: Sender: OAT; Receiver: OAT; Action: WAITING: Verify TDAFP_On ; Turbine_Driven_Aux_Feed_Pump_Auto_Start
307.884	Component: Sender: OAT; Receiver: PLANT; Component: TDAFP_On; Status: ACTIVE; Exp State: OFF; Observed State: NONE
308.413	Mental Belief: Sender: OAT; Receiver: OAT; Belief: Align_TDAFP_Flow_Path; Status: SUCCEED
308.942	Component: Sender: OAT; Receiver: OAT; Component: TDAFP_On; Status: FAILED; Exp State: OFF; Observed State: ON
311.06	New Alarm: Clear A_Tave_Hi_Dev
315.294	Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_NONE; Step_1
316.353	Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_1.2
320.588	ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify Median_Tave ; Check_Reactor_Coolant_System_Temperature - Check_for_Low_Taverage
321.117	Mental Belief: Sender: ODM; Receiver: ODM; Belief: Uncomplicated_Trip_Recovery; Status: SUCCEED
321.646	Parameter: Sender: ODM; Receiver: ODM; Parameter: Median_Tave; Status: FAILED; Exp State: <= 546.; Observed value: 548.111563553; Use of memorized info
323.234	Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_1.4
327.469	ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify Median_Tave ; Check_Reactor_Coolant_System_Temperature:_High_Tavg_-_Ensure_steam_dump_system_in_automatic
327.998	Parameter: Sender: ODM; Receiver: ODM; Parameter: Median_Tave; Status: SUCCEED; Exp State: > 548.; Observed value: 548.111563553; Use of memorized info
329.057	Parameter: Sender: ODM; Receiver: ODM; Parameter: RATE_Median_Tave; Status: SUCCEED; Exp State: <= -1.; Observed value: -2.26966371983; Use of memorized info
331.174	Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_2.1
335.409	ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify Median_Tave ; Check_Feed_Water_Status
335.938	Parameter: Sender: ODM; Receiver: ODM; Parameter: Median_Tave; Status: SUCCEED; Exp State: <= 554.; Observed value: 548.111563553; Use of memorized info
338.055	Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_2.2
342.29	ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify SG_A_FWRV_VPI ; Check_Feed_Water_Status_-_Isolate_Main_FWRVs
343.348	Parameter: Sender: ODM; Receiver: OAT; Parameter: SG_A_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: -999.
	Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_Align_MDAFP_Valves; Step_1
347.583	Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_A_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: -999.
348.641	Parameter: Sender: OAT; Receiver: ODM; Parameter: SG_A_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: 0.
349.171	Mental Belief: Sender: OAT; Receiver: OAT; Belief: MDAFP_Flowpath_Aligned; Status: SUCCEED
351.288	ACTION: Sender: OAT; Receiver: PLANT; Action: X_SG_A_MDAFW_Throttle ; Intended_Receiver:PLANT; Control_value:1. ; Align_MDAFP_Flowpath

352.346 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Uncomplicated_Trip_Recovery; Status: SUCCEED
353.405 Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_A_MDAFW_VPI; Status: ACTIVE; Exp State: >= 0.99; Observed value: -999.
353.934 Parameter: Sender: OAT; Receiver: OAT; Parameter: SG_A_MDAFW_VPI; Status: SUCCEED; Exp State: >= 0.99; Observed value: 1.
357.639 ACTION: Sender: OAT; Receiver: PLANT; Action: X_SG_B_MDAFW_Throttle ; Intended_Receiver:PLANT; Control_value:1. ; Align_MDAFP_Flowpath
359.757 Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_B_MDAFW_VPI; Status: ACTIVE; Exp State: >= 0.99; Observed value: -999.
360.286 Parameter: Sender: OAT; Receiver: OAT; Parameter: SG_B_MDAFW_VPI; Status: SUCCEED; Exp State: >= 0.99; Observed value: 1.
363.991 ACTION: Sender: OAT; Receiver: PLANT; Action: X_SG_C_MDAFW_Throttle ; Intended_Receiver:PLANT; Control_value:1. ; Align_MDAFP_Flowpath
366.108 Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_C_MDAFW_VPI; Status: ACTIVE; Exp State: >= 0.99; Observed value: -999.
366.637 Parameter: Sender: OAT; Receiver: OAT; Parameter: SG_C_MDAFW_VPI; Status: SUCCEED; Exp State: >= 0.99; Observed value: 1.
369.284 Procedure: Sender: OAT; Receiver: OAT; Procedure: MPBG_Align_TDAFP_Valves; Step_1
369.813 Mental Belief: Sender: OAT; Receiver: OAT; Belief: TDAFP_Flowpath_Aligned; Status: SUCCEED
371.93 ACTION: Sender: OAT; Receiver: PLANT; Action: X_SG_A_TDAFW_Throttle ; Intended_Receiver:PLANT; Control_value:1. ; Align_TDAFP_Flowpath
374.047 Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_A_TDAFW_VPI; Status: ACTIVE; Exp State: >= 0.99; Observed value: -999.
374.577 Parameter: Sender: OAT; Receiver: OAT; Parameter: SG_A_TDAFW_VPI; Status: SUCCEED; Exp State: >= 0.99; Observed value: 1.
378.282 ACTION: Sender: OAT; Receiver: PLANT; Action: X_SG_B_TDAFW_Throttle ; Intended_Receiver:PLANT; Control_value:1. ; Align_TDAFP_Flowpath
380.399 Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_B_TDAFW_VPI; Status: ACTIVE; Exp State: >= 0.99; Observed value: -999.
380.928 Parameter: Sender: OAT; Receiver: OAT; Parameter: SG_B_TDAFW_VPI; Status: SUCCEED; Exp State: >= 0.99; Observed value: 1.
383.575 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Uncomplicated_Trip_Recovery; Status: SUCCEED
384.633 ACTION: Sender: OAT; Receiver: PLANT; Action: X_SG_C_TDAFW_Throttle ; Intended_Receiver:PLANT; Control_value:1. ; Align_TDAFP_Flowpath
386.75 Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_C_TDAFW_VPI; Status: ACTIVE; Exp State: >= 0.99; Observed value: -999.
387.28 Parameter: Sender: OAT; Receiver: OAT; Parameter: SG_C_TDAFW_VPI; Status: SUCCEED; Exp State: >= 0.99; Observed value: 1.
389.926 Parameter: Sender: ODM; Receiver: ODM; Parameter: SG_A_FWRV_VPI; Status: SUCCEED; Exp State: <= 1.e-003; Observed value: 0.

395.748 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify SG_B_FWRV_VPI ; Check_Feed_Water_Status_-_Isolate_Main_FWRVs

396.807 Parameter: Sender: ODM; Receiver: OAT; Parameter: SG_B_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: -999.

401.041 Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_B_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: -999.

402.1 Parameter: Sender: OAT; Receiver: ODM; Parameter: SG_B_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: 0.

402.629 Parameter: Sender: ODM; Receiver: ODM; Parameter: SG_B_FWRV_VPI; Status: SUCCEED; Exp State: <= 1.e-003; Observed value: 0.

408.451 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify SG_C_FWRV_VPI ; Check_Feed_Water_Status_-_Isolate_Main_FWRVs

409.51 Parameter: Sender: ODM; Receiver: OAT; Parameter: SG_C_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: -999.

411.097 New Alarm: Actuation A_PZR_Level_Lo_Dev

413.744 Parameter: Sender: OAT; Receiver: PLANT; Parameter: SG_C_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: -999.

414.802 Parameter: Sender: OAT; Receiver: ODM; Parameter: SG_C_FWRV_VPI; Status: ACTIVE; Exp State: <= 1.e-003; Observed value: 0.

415.332 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Uncomplicated_Trip_Recovery; Status: SUCCEED

415.861 Parameter: Sender: ODM; Receiver: ODM; Parameter: SG_C_FWRV_VPI; Status: SUCCEED; Exp State: <= 1.e-003; Observed value: 0.

417.978 Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_2.3

422.212 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify Total_AFW_Flow ; Check_Feed_Water_Status_-_Total_Feed_Water_Flow

422.742 Parameter: Sender: ODM; Receiver: ODM; Parameter: Total_AFW_Flow; Status: SUCCEED; Exp State: >= 55.1; Observed value: 175.81093442; Use of memorized info

424.859 Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_3

425.388 Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_4.1a

429.622 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify PZR_Level ; Check_Pressuizer_Level_Control

430.681 Parameter: Sender: ODM; Receiver: OAT; Parameter: PZR_Level; Status: ACTIVE; Exp State: <= 0.12; Observed value: -999.

434.915 Parameter: Sender: OAT; Receiver: PLANT; Parameter: PZR_Level; Status: ACTIVE; Exp State: <= 0.12; Observed value: -999.

435.974 Parameter: Sender: OAT; Receiver: ODM; Parameter: PZR_Level; Status: ACTIVE; Exp State: <= 0.12; Observed value: 0.169598380742

436.503 Parameter: Sender: ODM; Receiver: ODM; Parameter: PZR_Level; Status: FAILED; Exp State: <= 0.12; Observed value: 0.169598380742

438.091 Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_4.2

442.325 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify Makeup_Flow ;

	Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
442.854	Parameter: Sender: ODM; Receiver: ODM; Parameter: PZR_Level; Status: FAILED; Exp State: >= 0.23; Observed value: 0.169598380742; Use of memorized info
445.501	ACTION: Sender: ODM; Receiver: OAT; Action: X_Letdown ; Intended_Receiver:OAT; Control_value:1. ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
449.735	ACTION: Sender: OAT; Receiver: PLANT; Action: X_Letdown ; Intended_Receiver:PLANT; Control_value:1. ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
450.793	ACTION: Sender: OAT; Receiver: ODM; Action: X_Letdown ; Intended_Receiver:ODM; Control_value:1. ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
451.323	Mental Belief: Sender: ODM; Receiver: ODM; Belief: Uncomplicated_Trip_Recovery; Status: SUCCEED
452.91	ACTION: Sender: ODM; Receiver: OAT; Action: X_Letdown ; Intended_Receiver:OAT; Control_value:-1. ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
457.145	ACTION: Sender: OAT; Receiver: PLANT; Action: X_Letdown ; Intended_Receiver:PLANT; Control_value:-1. ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
458.203	ACTION: Sender: OAT; Receiver: ODM; Action: X_Letdown ; Intended_Receiver:ODM; Control_value:-1. ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
459.262	Parameter: Sender: ODM; Receiver: OAT; Parameter: Makeup_Flow; Status: ACTIVE; Exp State: >= 0.; Observed value: -999.
463.496	Parameter: Sender: OAT; Receiver: PLANT; Parameter: Makeup_Flow; Status: ACTIVE; Exp State: >= 0.; Observed value: -999.
464.555	Parameter: Sender: OAT; Receiver: ODM; Parameter: Makeup_Flow; Status: ACTIVE; Exp State: >= 0.; Observed value: 1.94864565366
465.084	Parameter: Sender: ODM; Receiver: ODM; Parameter: Makeup_Flow; Status: SUCCEED; Exp State: >= 0.; Observed value: 1.94864565366
467.73	ACTION: Sender: ODM; Receiver: OAT; Action: X_PZR_Level_Setpoint ; Intended_Receiver:OAT; Control_value:0.22 ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
471.964	ACTION: Sender: OAT; Receiver: PLANT; Action: X_PZR_Level_Setpoint ; Intended_Receiver:PLANT; Control_value:0.22 ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
473.023	ACTION: Sender: OAT; Receiver: ODM; Action: X_PZR_Level_Setpoint ; Intended_Receiver:ODM; Control_value:0.22 ; Check_Pressuizer_Level_Control_-_Verify_CVCS_in_Service
474.081	Parameter: Sender: ODM; Receiver: OAT; Parameter: PZR_Level_Setpoint; Status: ACTIVE; Exp State: BETWEEN 0.21 and 0.23; Observed value: -999.
477.786	New Alarm: Actuation A_Tave_Lo_Dev
478.316	Parameter: Sender: OAT; Receiver: PLANT; Parameter: PZR_Level_Setpoint; Status: ACTIVE; Exp State: BETWEEN 0.21 and 0.23; Observed value: -999.
479.374	Parameter: Sender: OAT; Receiver: ODM; Parameter: PZR_Level_Setpoint; Status: ACTIVE; Exp State: BETWEEN 0.21 and 0.23; Observed value: 0.22
479.903	Mental Belief: Sender: ODM; Receiver: ODM; Belief: Alarm_Lo_Tave_Deviation; Status: SUCCEED
480.433	Parameter: Sender: ODM; Receiver: ODM; Parameter: PZR_Level_Setpoint; Status: SUCCEED; Exp State: BETWEEN 0.21 and 0.23; Observed value: 0.22

480.962 Procedure: Sender: ODM; Receiver: ODM; Procedure: MPBG_ARP_Low_Tave_Deviation; Step_1
481.491 AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: Tave-Tref
482.55 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Uncomplicated_Trip_Recovery; Status: SUCCEED
483.079 AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: Median_Tave
483.608 New Alarm: Clear A_SG_B_Hi_Pressure
484.138 AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: RATE_Median_Tave
484.667 New Alarm: Clear A_SG_A_Hi_Pressure
487.313 Mental Belief: Sender: ODM; Receiver: ODM; Belief: Alarm_Low_Pressurizer_Pressure; Status: SUCCEED
487.843 Procedure: Sender: ODM; Receiver: ODM; Procedure: ES_0.1; Step_4.3
488.372 Procedure: Sender: ODM; Receiver: ODM; Procedure: MPBG_ARP_PZR_Low_Pressure; Step_1
488.901 AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: PZR_Pressure
489.96 AddScannedParameter: Sender: ODM; Receiver: ODM; Parameter: RATE_PZR_Pressure
492.077 New Alarm: Clear A_SG_C_Hi_Pressure
495.252 ACTION: Sender: ODM; Receiver: ODM; Action: WAITING: Verify PZR_Level ; Check_Pressuizer_Level_Control_-_Maintain_Lpzr_at_22%
496.311 Parameter: Sender: ODM; Receiver: OAT; Parameter: PZR_Level; Status: ACTIVE; Exp State: <= 0.21; Observed value: -999.
500.545 Parameter: Sender: OAT; Receiver: PLANT; Parameter: PZR_Level; Status: ACTIVE; Exp State: <= 0.21; Observed value: -999.
501.604 Parameter: Sender: OAT; Receiver: ODM; Parameter: PZR_Level; Status: ACTIVE; Exp State: <= 0.21; Observed value: 0.155612294108
502.133 Parameter: Sender: ODM; Receiver: ODM; Parameter: PZR_Level; Status: SUCCEED; Exp State: <= 0.21; Observed value: 0.155612294108

Appendix F – Knowledge-Based Actions

Appendix F provides a listing of the knowledge-based actions included in the three loop pressurized water reactor ADS-IDAC model. These knowledge-based actions are sufficient to guide a non-proceduralized operator response to a steam generator tube rupture and a loss of reactor coolant accident event. Individual actions are activated when the operator perceives a deficiency in a system function associated with mass, energy, or momentum control. The specific format used for this input is described in Appendix K, the ADS-IDAC input manual.

```
number_of_diagnoses     5

diagnosis_name    Mass_Imbalance_PZR_Low
diagnosis_priority      5
reset_delay_time        100.0 1.0   1.0
number_of_actions       5

      action_name_1     X_PZR_PORV
      action_priority   2
      action_type       3074
      control_input     0.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 1
            PZR_Pressure      3045  0.0   1.0   2.0   3045  3009  2290.0      0.0   3045

      action_name_2     X_Letdown
      action_priority   1
      action_type       3074
      control_input     0.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 1
            PZR_Level   3045  0.0   1.0   2.0   3045  3009  0.15  0.0   3045

      action_name_3     X_SIAS
      action_priority   3
      action_type       3074
      control_input     1.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 3
            PZR_Level         3045  0.0   1.0   2.0   3045  3009  0.10        0.0   3003
            PZR_Pressure      3045  0.0   1.0   2.0   3045  3009  2150.0      0.0   3003
            Safety_Injection  3045  0.0   1.0   2.0   3023  3045
```

```
      action_name_4     X_HPI_Pump_A
      action_priority   4
      action_type       3074
      control_input     1.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 2
            A_HPI_Pump_On     3045  1.0   3.0   2.0   3023  3003
            PZR_Level         3045  0.0   1.0   2.0   3045  3009  0.10  0.0   3045

      action_name_5     X_HPI_Pump_B
      action_priority   5
      action_type       3074
      control_input     1.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 2
            B_HPI_Pump_On     3045  1.0   3.0   2.0   3023  3045
            PZR_Level         3045  0.0   1.0   2.0   3045  3009  0.10  0.0   3045
```

diagnosis_name   Mass_Imbalance_PZR_High
diagnosis_priority   5
reset_delay_time   200.0   1.0   1.0
number_of_actions   2

```
      action_name_1     X_HPI_Pump_A
      action_priority   2
      action_type       3074
      control_input     0.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 3
            A_HPI_Pump_On     3045  1.0   3.0   2.0   3022  3003
            Min_Sub_Cooling   3045  1.0   3.0   2.0   3045  3007  10.0  0.0   3045
            PZR_Level         3045  0.0   1.0   2.0   3045  3007  0.20  0.0   3045
```

```
    action_name_2     X_HPI_Pump_B
    action_priority   2
    action_type       3074
    control_input     0.0
    lower_limit       0.0
    upper_limit       1.0
    number_of_prerequisites 3
        B_HPI_Pump_On     3045  1.0   3.0   2.0   3022  3045
        Min_Sub_Cooling   3045  1.0   3.0   2.0   3045  3007  10.0  0.0   3045
        PZR_Level         3045  0.0   1.0   2.0   3045  3007  0.20  0.0   3045
```

diagnosis_name   Mass_Imbalance_SG_A_High
diagnosis_priority   5
reset_delay_time   200.0   1.0   1.0
number_of_actions   3

```
    action_name_1     X_PZR_Spray_Valve
    action_priority   2
    action_type       3074
    control_input     1.0
    lower_limit       0.0
    upper_limit       1.0
    number_of_prerequisites 2
        Mental_Belief           Possible_SG_Tube_Rupture            3021  3003
        Parameter_Difference    PZR_Pressure       SGTR_Pressure     3002  1.0   3.0   2.0   50.0  3045

    action_name_2     X_PZR_Spray_Valve
    action_priority   2
    action_type       3074
    control_input     0.0
    lower_limit       0.0
    upper_limit       1.0
    number_of_prerequisites 2
        Mental_Belief           Possible_SG_Tube_Rupture      3021  3003
```

```
            Parameter_Difference    SGTR_Pressure     PZR_Pressure            3002  1.0   3.0   2.0   -25.0
                  3045

      action_name_3     X_PZR_Heaters
      action_priority   2
      action_type       3074
      control_input     0.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 1
            Mental_Belief           Possible_SG_Tube_Rupture      3021  3045
```

diagnosis_name   Energy_Imbalance_PZR_Low
diagnosis_priority   5
reset_delay_time   200.0   1.0   1.0
number_of_actions   10

```
      action_name_1     X_PZR_PORV
      action_priority   1
      action_type       3074
      control_input     0.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 1
            PZR_Pressure      3045  0.0   1.0   2.0   3045  3009  2290.0      0.0   3045


      action_name_2     X_SIAS
      action_priority   2
      action_type       3074
      control_input     1.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 3
            PZR_Level         3045  0.0   1.0   2.0   3045  3009  0.10  0.0   3002
```

```
      Min_Sub_Cooling   3045  1.0   3.0   2.0   3045  3009  10.0  0.0   3003
      PZR_Pressure      3045  0.0   1.0   2.0   3045  3009  1900.0      0.0   3003
      Safety_Injection  3045  0.0   1.0   2.0   3023  3045

action_name_3     X_HPI_Pump_A
action_priority   2
action_type       3074
control_input     1.0
lower_limit       0.0
upper_limit       1.0
number_of_prerequisites 2
      A_HPI_Pump_On     3045  1.0   3.0   2.0   3023  3003
      PZR_Level         3045  0.0   1.0   2.0   3045  3009  0.10  0.0   3002
      Min_Sub_Cooling   3045  1.0   3.0   2.0   3045  3009  10.0  0.0   3045


action_name_4     X_HPI_Pump_B
action_priority   2
action_type       3074
control_input     1.0
lower_limit       0.0
upper_limit       1.0
number_of_prerequisites 2
      B_HPI_Pump_On     3045  1.0   3.0   2.0   3023  3045
      PZR_Level         3045  0.0   1.0   2.0   3045  3009  0.10  0.0   3002
      Min_Sub_Cooling   3045  1.0   3.0   2.0   3045  3009  10.0  0.0   3045


action_name_5     X_SG_A_PORV_Setpoint
action_priority   3
action_type       3139
control_input     SG_A_Pressure
lower_limit       400.0
upper_limit       1025.0
number_of_prerequisites 5
      Mental_Belief           Possible_SG_Tube_Rupture      3021  3045
```

```
      Min_Sub_Cooling         3002  1.0   3.0   2.0   3045  3009  30.0  0.0   3045
      FLAG_SGTR_SG_A          3045  0.0   1.0   2.0   3023  3045
      SG_A_PORV_Setpoint      3002  1.0   3.0   2.0   3045  3007  1000.0      0.0   3045
      SG_A_Pressure           3002  1.0   3.0   2.0   3045  3032  400.0 1025.0      3045

action_name_6     X_SG_B_PORV_Setpoint
action_priority   3
action_type       3139
control_input     SG_B_Pressure
lower_limit       400.0
upper_limit       1025.0
number_of_prerequisites 5
      Mental_Belief           Possible_SG_Tube_Rupture      3021  3045
      Min_Sub_Cooling         3045  1.0   3.0   2.0   3045  3009  30.0  0.0   3045
      FLAG_SGTR_SG_B          3045  0.0   1.0   2.0   3023  3045
      SG_B_PORV_Setpoint      3002  1.0   3.0   2.0   3045  3007  1000.0      0.0   3045
      SG_B_Pressure           3002  1.0   3.0   2.0   3045  3032  400.0 1025.0      3045

action_name_7     X_SG_C_PORV_Setpoint
action_priority   3
action_type       3139
control_input     SG_C_Pressure
lower_limit       400.0
upper_limit       1025.0
number_of_prerequisites 5
      Mental_Belief           Possible_SG_Tube_Rupture      3021  3045
      Min_Sub_Cooling         3045  1.0   3.0   2.0   3045  3009  30.0  0.0   3045
      FLAG_SGTR_SG_C          3045  0.0   1.0   2.0   3023  3045
      SG_C_PORV_Setpoint      3002  1.0   3.0   2.0   3045  3007  1000.0      0.0   3045
      SG_C_Pressure           3002  1.0   3.0   2.0   3045  3032  400.0 1025.0      3045


action_name_8     X_SG_A_PORV_Setpoint
action_priority   4
action_type       3112
control_input     -25.0
```

```
      lower_limit       400.0
      upper_limit       1025.0
      number_of_prerequisites 4
            SG_A_PORV_VPI          3002  0.0   1.0   2.0   3045  3009  0.5   0.0   3045
            Mental_Belief          Possible_SG_Tube_Rupture    3021  3045
            Min_Sub_Cooling        3045  1.0   3.0   2.0   3045  3009  20.0  0.0   3045
            FLAG_SGTR_SG_A         3045  0.0   1.0   2.0   3023  3045

      action_name_9     X_SG_B_PORV_Setpoint
      action_priority   4
      action_type       3112
      control_input     -25.0
      lower_limit       400.0
      upper_limit       1025.0
      number_of_prerequisites 4
            SG_B_PORV_VPI          3002  0.0   1.0   2.0   3045  3009  0.5   0.0   3045
            Mental_Belief          Possible_SG_Tube_Rupture    3021  3045
            Min_Sub_Cooling        3045  1.0   3.0   2.0   3045  3009  20.0  0.0   3045
            FLAG_SGTR_SG_B         3045  0.0   1.0   2.0   3023  3045

      action_name_10    X_SG_C_PORV_Setpoint
      action_priority   4
      action_type       3112
      control_input     -25.0
      lower_limit       400.0
      upper_limit       1025.0
      number_of_prerequisites 4
            SG_C_PORV_VPI    3002  0.0   1.0   2.0   3045  3009  0.5   0.0   3045
            Mental_Belief    Possible_SG_Tube_Rupture    3021  3045
            Min_Sub_Cooling  3045  1.0   3.0   2.0   3045  3009  20.0  0.0   3045
            FLAG_SGTR_SG_C   3045  0.0   1.0   2.0   3023  3045
```

diagnosis_name   Energy_Imbalance_RCS_Low
diagnosis_priority   5
reset_delay_time   200.0   1.0   1.0

```
number_of_actions       3

      action_name_1     X_SG_A_Atmos_PORV
      action_priority   1
      action_type       3074
      control_input     0.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 3
            Mental_Belief          Goal_Reduce_RCS_Temperature   3045  3045
            SG_A_PORV_VPI          3002  0.0   1.0   2.0   3045  3007  0.001       0.0   3045
            SG_A_Pressure          3002  1.0   3.0   2.0   3045  3007  1025.0      0.0   3045

      action_name_2     X_SG_B_Atmos_PORV
      action_priority   1
      action_type       3074
      control_input     0.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 3
            Mental_Belief          Goal_Reduce_RCS_Temperature   3045  3045
            SG_B_PORV_VPI          3002  0.0   1.0   2.0   3045  3007  0.001       0.0   3045
            SG_B_Pressure          3002  1.0   3.0   2.0   3045  3007  1025.0      0.0   3045

      action_name_3     X_SG_C_Atmos_PORV
      action_priority   1
      action_type       3074
      control_input     0.0
      lower_limit       0.0
      upper_limit       1.0
      number_of_prerequisites 3
            Mental_Belief          Goal_Reduce_RCS_Temperature   3045  3045
            SG_C_PORV_VPI          3002  0.0   1.0   2.0   3045  3007  0.001       0.0   3045
            SG_C_Pressure          3002  1.0   3.0   2.0   3045  3007  1025.0      0.0   3045
```

(this page intentionally blank)

Appendix G – LOFW Scenario: Predicted Branching Events

Appendix G provides a summary of the branching events that were predicted to be relevant to the Halden Empirical HRA Study Loss of Feedwater scenarios. For each predicted branching event, the associated source of crew-to-crew variability is described in addition to the impact that the crew action would have on the nuclear plant. These branching events were identified without knowledge of the actual crew performance during the experimental study. Appendix H provides a comparison of these predicted events and the actual observed operator behaviors.

Branching Event	Scenario Applicability	Associated Crew Behavior Variability	Plant Impact
Early vs. Late Reactor Trip	Base and Complex	Awareness of degrading plant conditions. Crew operating philosophy.	An early manual reactor trip preserves a greater secondary water inventory in the steam generators compared to an automatic reactor trip on low SG water level. Because the feed and bleed initiation criteria is based on SG water level, a greater inventory increases the time available until feed and bleed must be initiated.
Early vs. Late Detection of Low Auxiliary Feedwater Condition	Base and Complex	Awareness of degrading plant conditions. The EOPs prompt the operators to verify the adequacy of the secondary heat sink (ES-0.1, Step 6). However, the operators may detect the low auxiliary feedwater flow condition before this step through self-initiated control panel monitoring activities.	Early detection of a low auxiliary feedwater flow condition results in an earlier transition to FRG H.1, reduces the time until the RCPs are tripped, and minimizes the likelihood of a late initiation of feed and bleed.
Time Required to Transition from ES-0.1 to FRG H.1	Base and Complex	Crew ability to shift to a new high level goal and implement a new procedure	A shorter transition time to FRG H.1 results in an earlier transition to FRG H.1 reduces the time until the RCPs are tripped and minimizes the likelihood of a late initiation of feed and bleed.
FRG H.1, Briefing Hold 1	Base and Complex	Procedure execution speed and crew communication tendencies	Briefing Hold 1 occurs just prior to FRG H.1 Step 1. Because the Briefing Hold 1 time must elapse before the FRG H.1 actions can be initiated, a longer hold time delays tripping of the reactor coolant pumps (and reduces the time available for initiation of feed and bleed).
Skip FRG H.1, Step 3.0 (Trip Reactor Coolant Pumps)	Base and Complex	Procedural adherence tendency. Knowledge and recognition of the importance of early tripping of the RCPs	Tripping the reactor coolant pumps minimizes the reactor coolant system heat addition rate. This, in turn, minimizes the steaming rate from the SGs and increases the time available until feed and bleed must be initiated.

Branching Event	Scenario Applicability	Associated Crew Behavior Variability	Plant Impact
Time Based Transition to Feed & Bleed (Watchdog Timer)	Base and Complex	Variations in interpretation and implementation of the SG level transition criteria for feed and bleed initiation	For the Base case scenarios, the watchdog timer is used to force an early transition to feed and bleed (i.e., the operators initiate feed and bleed when the watchdog timer expires, regardless of SG wide range level). Because all SG wide level instruments are functional in the Base case, feed and bleed will also be initiated when two SG wide range indicators reach 12% (regardless of the watchdog timer setting). During the complex case scenarios, two of the three wide range SG level instruments are stuck at a value above the feed and bleed transition criteria. Consequently, the procedural requirement for feed and bleed initiation based on SG level will never be satisfied. The watchdog timer is used to trigger the transition to feed and bleed (i.e., the operators initiate feed and bleed when the watchdog timer expires, regardless of the actual or perceived SG wide range level).
FRG H.1, Briefing Hold 2	Complex	Procedure execution speed and crew communication tendencies (e.g., conduct of crew meetings)	Briefing Hold 2 occurs just prior to FRG H.1 Step 7. Because the Briefing Hold 2 time must elapse before actions to align the condensate system can be initiated, a longer hold time delays the start of secondary depressurization. Delaying the start of depressurization reduces the time available to restore feed from the condensate before feed and bleed is initiated.
Target pressure value during reactor coolant system depressurization in FRG H.1 Step 7.a	Complex	Variability in selecting target values for dynamic parameters.	Too high of a target value may result in the inability to block the low pressure safety injection signal. Too low of a target value can result in an inadvertent safety injection if pressure is reduced below the low reactor coolant system pressure safety injection setpoint.

Branching Event	Scenario Applicability	Associated Crew Behavior Variability	Plant Impact
Failure to block safety injection actuation signals in FRG-H.1, Step 7.b	Complex	Procedural adherence tendency. Knowledge and recognition of the importance of preventing an inadvertent safety injection actuation while attempting main feedwater recovery	The failure to block these safety actuation signals may result in an inadvertent emergency core cooling actuation during subsequent recovery steps. This would result in an automatic isolation of the main feedwater system and further complicate and delay the plant recovery
Secondary cooldown rate (FRG H.1, Step 7.c)	Complex	Variability in choosing between the competing goals of minimizing the time required to depressurize the secondary and avoiding an excessive depressurization rate.	An excessive depressurization time delays the recovery of feed water from the condensate system. An excessive depressurization rate causes control difficulties and can result in either an inadvertent safety injection (due to low reactor coolant system pressure) or a main steam isolation signal.
FRG H.1, Briefing Hold 3	Base and Complex	Procedure execution speed and crew communication tendencies.	Briefing Hold 3 occurs just prior to FRG H.1 Step 10. Because Briefing Hold 3 must elapse before feed and bleed core cooling can be initiated, a longer hold time delays the transition from steam generator heat removal to feed and bleed. If the hold time is excessive, secondary SG inventory may be fully depleted prior restoration of effective core cooling.
Insufficient Bleed PORV Bleed Path (FRG H.1 Step 15.b)	Base and Complex	Procedural adherence tendency. Resolution of conflicting goals of minimizing loss of reactor coolant inventory and ensuring sufficient core cooling. Variation in control input.	In order to adequately cool the reactor core, the reactor coolant bleed mass flow rate through the pressurizer PORV valves must exceed the core boil off rate. The failure to open all available pressurizer PORV valves limits the bleed mass flow rate and could result in insufficient core cooling.

Appendix H – LOFW Scenario: Crew Performance Summary

Appendix H provides a comparison of the crew-to-crew variations predicted with the ADS-IDAC model and the actual observed crew behaviors from the Halden FRESH simulator exercises. For each predicted branching event, a comparison to the crew data for the base and complex scenarios is made to determine if the behavior was actually observed.

Predicted Branching Event	Observed During Base Scenario	Observed During Complex Scenario?	Comments
Early vs. Late Reactor Trip	Yes Ten of fourteen crews manually tripped the reactor early. Only four crews (I, J, K, and L) delayed and received an automatic trip	Yes Based on a review of the data, it is likely that three crews tripped the reactor early (B, D, and E) while the remaining crews delayed action and received an automatic reactor trip.	The reversal in predominance of manual tripping between the base and complex scenarios would indicate that the scenario structure influenced the decision to trip early. Because condensate flow remained available during the complex scenario, it is possible crews attempted to recover main feed rather than shutting the plant down. In the base case, the complete loss of condensate flow would make restoration of feed highly unlikely before a reactor trip was automatically generated – in this case the operators may have better anticipated the inevitable consequences of the event
Early vs. Late Detection of Low Auxiliary Feedwater Condition	Unknown	Unknown	Although it is likely that crews exhibited variance in their detection time for the loss of AFW condition, there is insufficient information in the simulator log data to support an assessment for this behavior.
Time Required to Transition from ES-0.1 to FRG H.1	Unknown	Unknown	Although it is likely that crews exhibited variance in the time required to transition from ES-0.1 to FR-H.1, there is insufficient information in the simulator log data to support an assessment for this behavior.
FRG H.1, Briefing Hold #1	Yes	Yes	Due to the inability to explicitly determine the crew timing for detection of the loss of AFW condition and the transition to FR-H.1, all timing variability for initiating FR-H.1 was lumped into Briefing Hold #1. Despite predicting this as a source of crew-to-crew variability, the initial prediction underestimated both the length of this hold point and the variance among crews.

Predicted Branching Event	Observed During Base Scenario	Observed During Complex Scenario?	Comments
Skip FRG H.1, Step 3.0 (Trip Reactor Coolant Pumps)	No However, Crew L tripped two reactor coolant pumps prior to starting FR-H.1 to minimize heat input to the RCS. The crew took a relatively long time to transfer to procedure FR-H.1 but did trip the third reactor coolant pump as directed by the procedure (approximately 13 minutes later). Although the actual crew behavior is different from that predicted by ADS-IDAC, the impact on the reactor plant is similar.	No However, Crew B took a relatively long time to transfer to FR-H.1 and initiated emergency core cooling prior to starting procedure FR-H.1. Although this crew did not actually skip the FR-H.1 step to trip reactor coolant pumps, the net impact on the plant was similar to a crew that transitioned to FR-H.1 in a more timely manner but failed to trip the reactor coolant pumps in accordance with the procedure.	The background document for the loss of secondary heat sink recovery guideline assumes a best estimate prediction that operators would trip the reactor coolant pumps within five minutes of the reactor trip. In both the base and complex scenario, no crew stopped the reactor coolant pumps within this time frame. The quickest time to stop the pumps was approximately six minutes (Crew N, Base case). Curiously, Crew N was among the slowest crews to trip the reactor coolant pumps during the complex case by delaying this action for approximately 24 minutes. The average time was roughly 10 minutes in the base case and 13 minutes for the complex case.
Time Based Transition to Feed & Bleed (Watchdog Timer)	See Comment Three crews initiated feed bleed prior to reaching the required SG water level criteria. Crews B, E, F, and H all initiated feed and bleed with wide range SG water levels significantly above 12%.	See Comment	For the base case, the watchdog timer was used to force an early transition to feed and bleed decay heat removal. For the complex case, the watchdog timer forced the transition to feed and bleed (since the requirement of at least two SGs less than 12% level could not be met due to instrument biasing). Although crews may not have explicitly based the transition to feed and bleed decay heat removal based on time criterion, this approach was useful for modeling the crew timing variability in initiating feed and bleed cooling.

Predicted Branching Event	Observed During Base Scenario	Observed During Complex Scenario?	Comments
FRG H.1, Briefing Hold #2	Not Applicable	Yes	Briefing Hold #2 (prior to RCS depressurization in Step 7 of FR-H.1) is not applicable to the base case since the condensate system was unavailable. Timing variability for this was observed during the complex case. Despite predicting this as a source of crew-to-crew variability, the initial prediction underestimated both the length of this hold point and the variance among crews.
Target pressure value during reactor coolant system depressurization in FRG H.1 Step 7.a	Not Observed	Not Observed	Although no crew inadvertently initiated a safety injection during RCS depressurization per step 7.a of FR-H.1,

Predicted Branching Event	Observed During Base Scenario	Observed During Complex Scenario?	Comments
Failure to block safety injection actuation signals in FRG-H.1, Step 7.b	Not Applicable	Yes Several crews failed to block the low pressure and/or high steam flow safety injection actuation as required by procedure. Crews A, E, and H failed to reduce pressure below the P11 permissive setpoint and were apparently unable to reset either of the safety injection blocks. Crews B, C, and G activated the P11 permissive during the scenario, but failed to block either the low pressure or high steam flow safety injection. Crews D and K successfully blocked the high steam flow signal, but failed to block the low pressure safety injection signal. In summary, of the fourteen crews, only six (crews F, I, J, L, M, and N) appear to have properly executed procedure steps 7.a and 7.b.	Although failure to properly perform this step could lead to an inadvertent safety actuation later in the procedure, no crew appears to have inadvertently actuated safety injection due to the failure to perform this step action. Only one crew appeared to actuate an inadvertent safety injection (Crew F), but this was due to a high differential steam generator condition rather than a high steam flow.
Secondary cooldown rate (FRG H.1, Step 7.c)	Not Applicable	Not Specifically Observed, though Crew F triggered an automatic safety injection actuation during depressurization due to developing a high differential pressure between SGs. This condition is approximately equivalent to the predicted branching event for the ADS-IDAC model.	This branching event highlights a difference between the FRESH simulator and the ADS-IDAC plant model. The FRESH simulator includes a high differential SG pressure safety injection when the difference between the pressure in two or more SGs exceeds roughly 100 psi. The reference plant model for ADS-IDAC does not include this trip – instead the safety injection signal is based on a high rate of depressurization. In this scenario, the activation of the high differential safety injection actuation is roughly equivalent to

Predicted Branching Event	Observed During Base Scenario	Observed During Complex Scenario?	Comments
FRG H.1, Briefing Hold #3	Unknown	Unknown	Although it is likely that crews exhibited variance in the time taken to initiate feed and bleed cooling, there is insufficient information in the simulator log data to support an assessment for this behavior. Of particular note is the longer than anticipated delay in establishing a PORV bleed path once high pressure injection flow was initiated.
Insufficient Bleed PORV Bleed Path (FRG H.1 Step 15.b)	Partially Two crews (D and F) had significant delays between initiation of high pressure injection and opening of a PORV bleed path. The average delay between injection and opening a bleed path was roughly 2-3 minutes. Crew D had a 5 minute delay while crew F had a 7 minute delay	Partially Of the 7 crews that successfully aligned a PORV bleed path, two crews (B and F) had an excessive delay between initiation of injection flow and opening of a bleed path. For the crews that established feed and bleed, the average delay between injection and opening a bleed path was roughly 5 minutes. Crew B had a 14 minute delay while crew F had a 7 minute delay. It should be noted that Crew F inadvertently actuated the feeding due to developing a high differential pressure between the SGs during secondary depressurization	Although no crew used only one or two pressurizer PORVs for feed and bleed, several crews exhibited extremely long delay times between initiation of high pressure injection and opening of the pressurizer PORVs. This condition could be interpreted as the failure to align an adequate bleed path from the RCS. It should also be noted that a scenario complication for the base case that was not included in the original scenario description resulted in the PORV fails to fail closed once RCS pressure decreased below ~2000 psi.

Appendix I – LOFW Base Case Comparison

Appendix I provides a series of graphs that compare the performance of control room crews at the Halden FRESH simulator facility during the base case loss of feedwater event to the results of the re-calibrated ADS-IDAC model. Although the ADS-IDAC results were obtained after the loss of feedwater empirical study data was reviewed and the ADS-IDAC knowledge base adjusted, this comparison demonstrates that the model is capable of representing a wide range of crew-to-crew variabilities with a relatively small number of branching events.

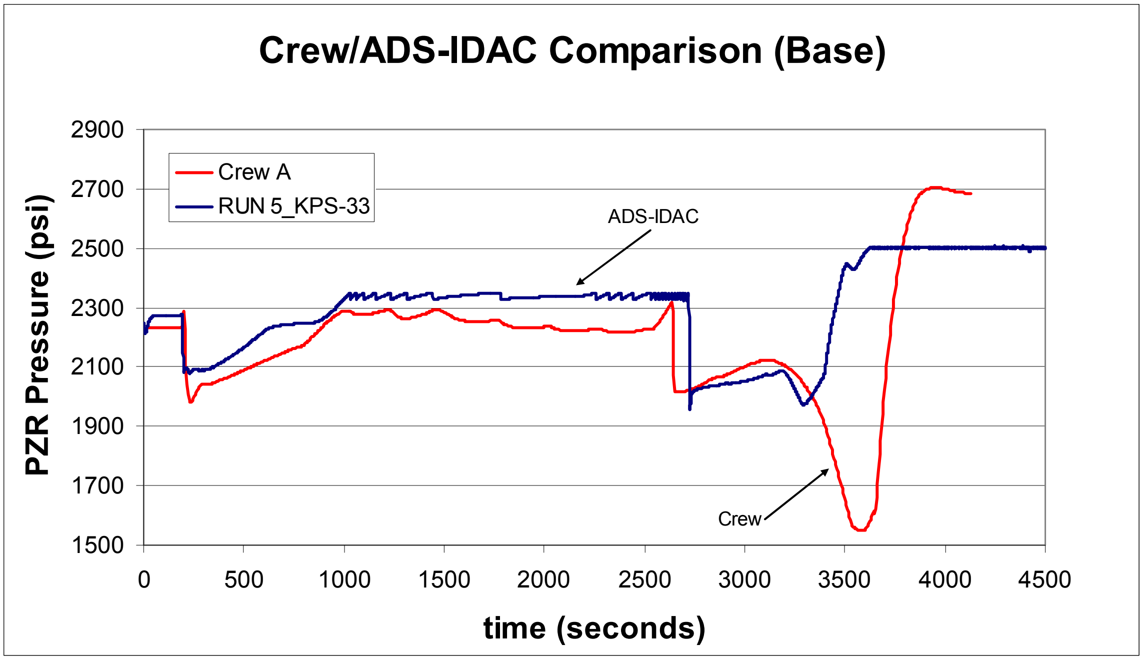


Figure 65 - Crew A (Base LOFW) Pressurizer Pressure

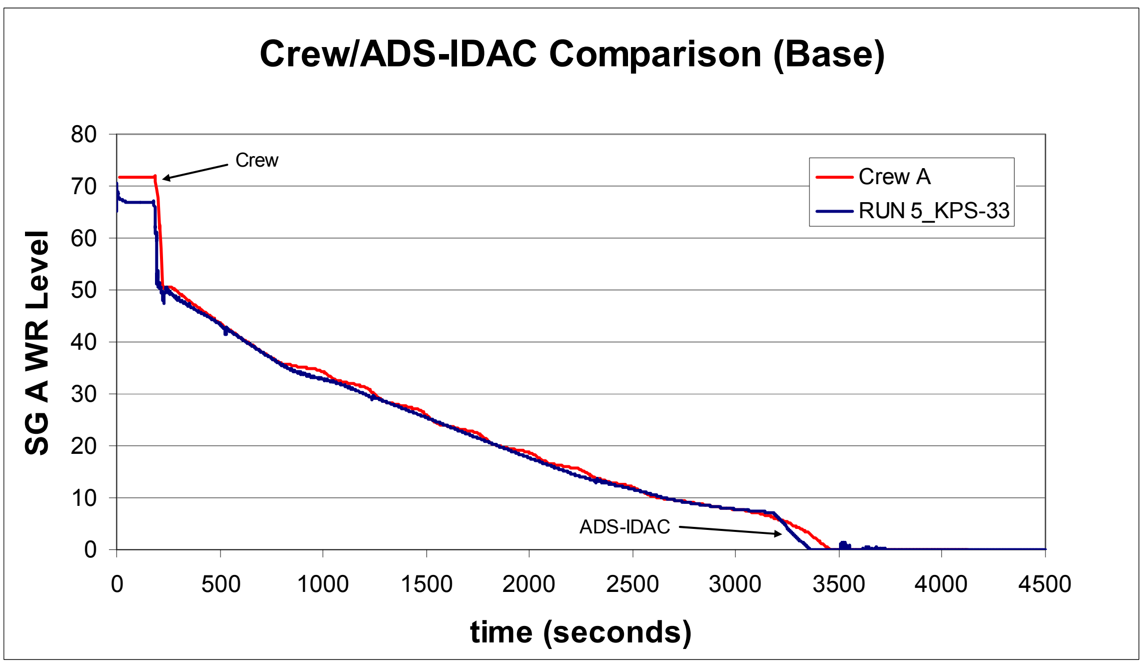


Figure 66 - Crew A (Base LOFW) SG A Wide Range Water Level

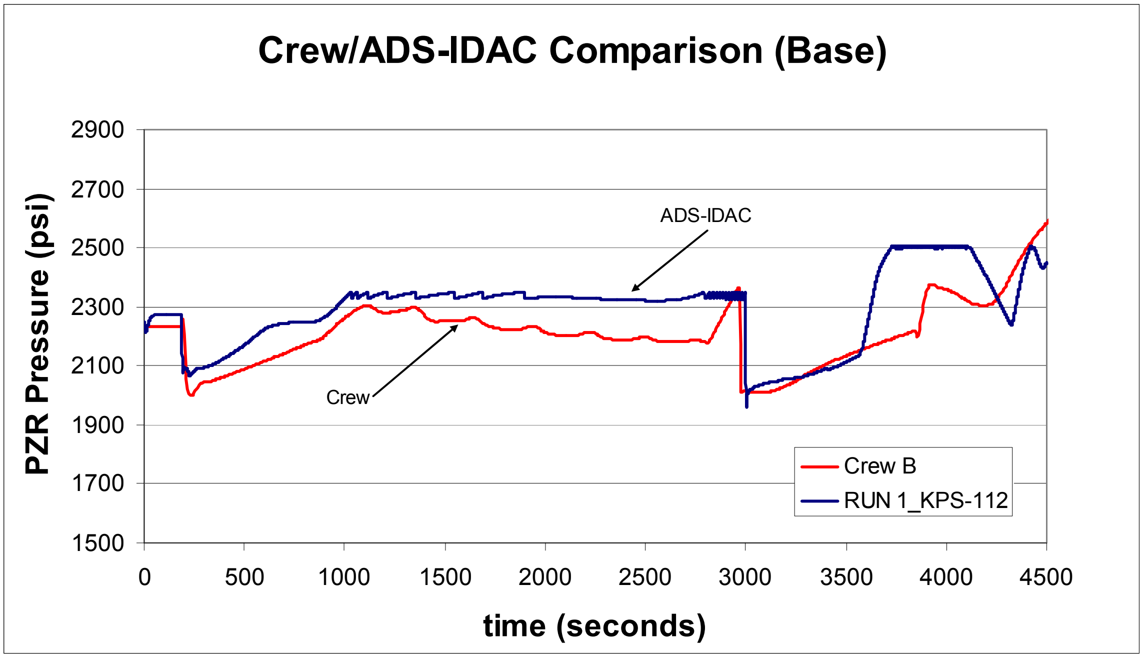


Figure 67 - Crew B (Base LOFW) Pressurizer Pressure

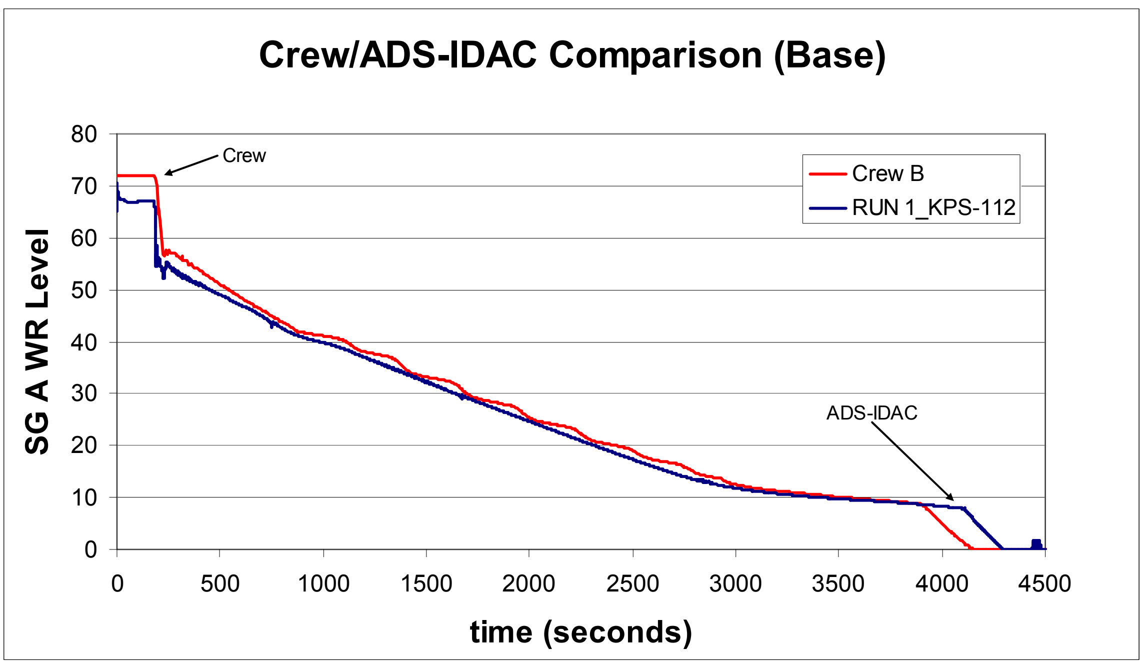


Figure 68 - Crew B (Base LOFW) SG A Wide Range Water Level

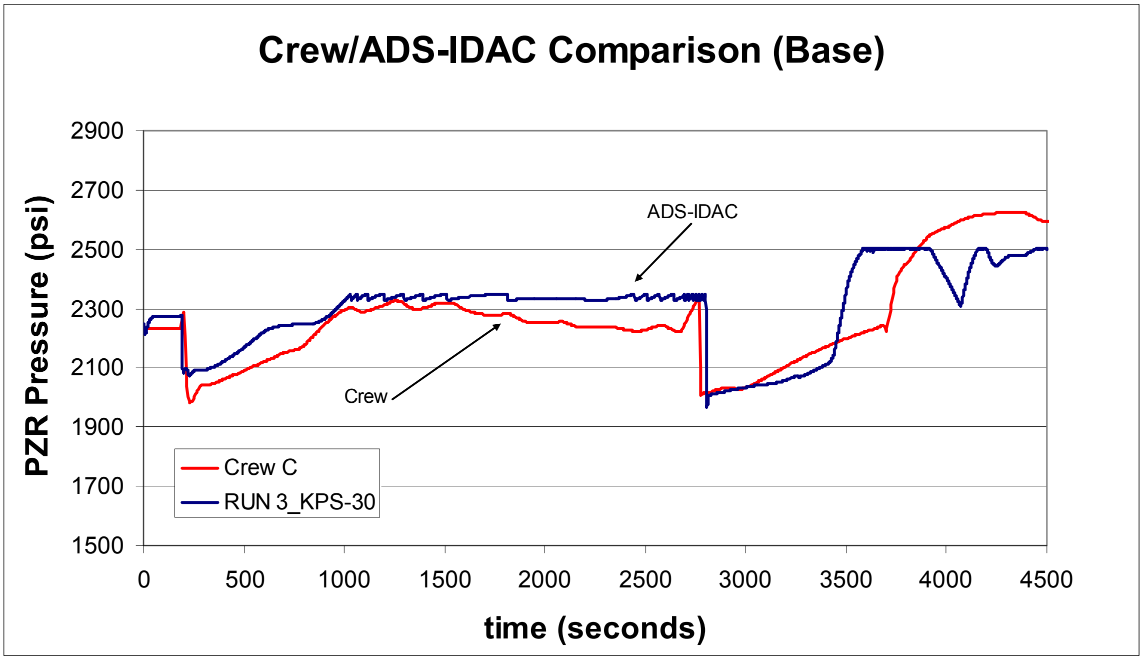


Figure 69 - Crew C (Base LOFW) Pressurizer Pressure

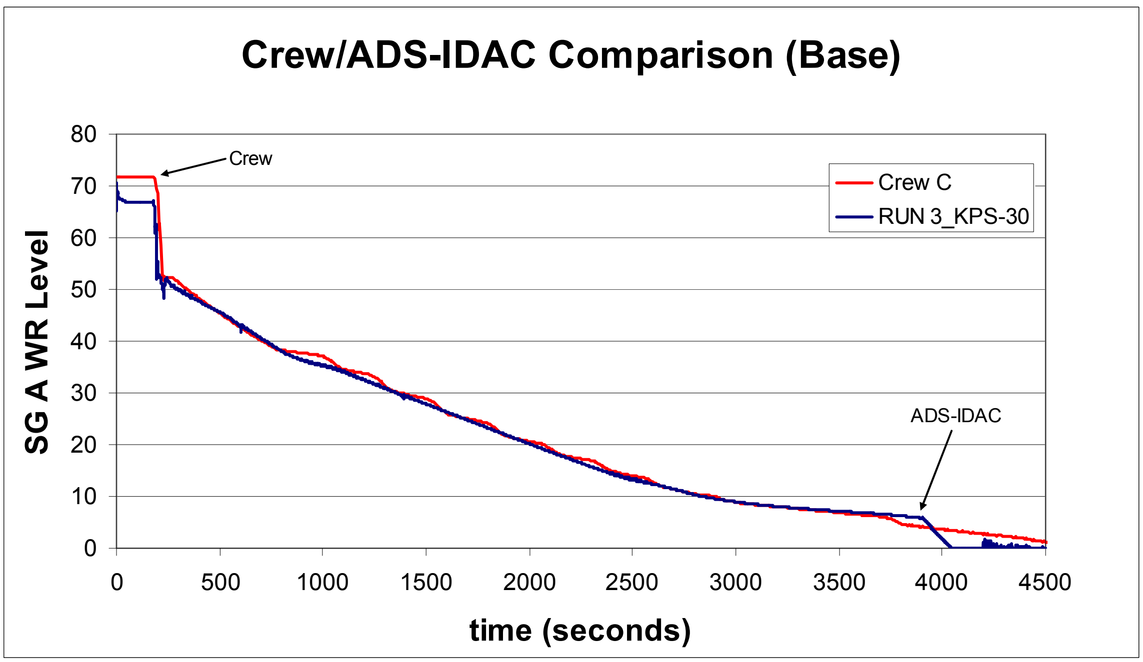


Figure 70 - Crew C (Base LOFW) SG A Wide Range Water Level

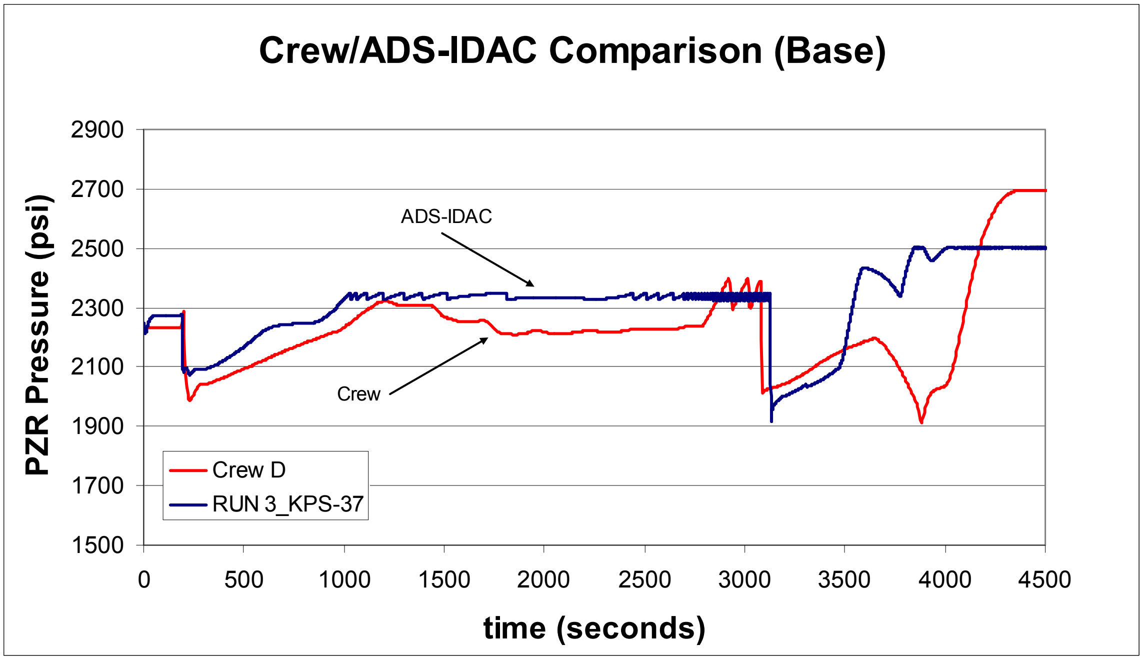


Figure 71 - Crew D (Base LOFW) Pressurizer Pressure

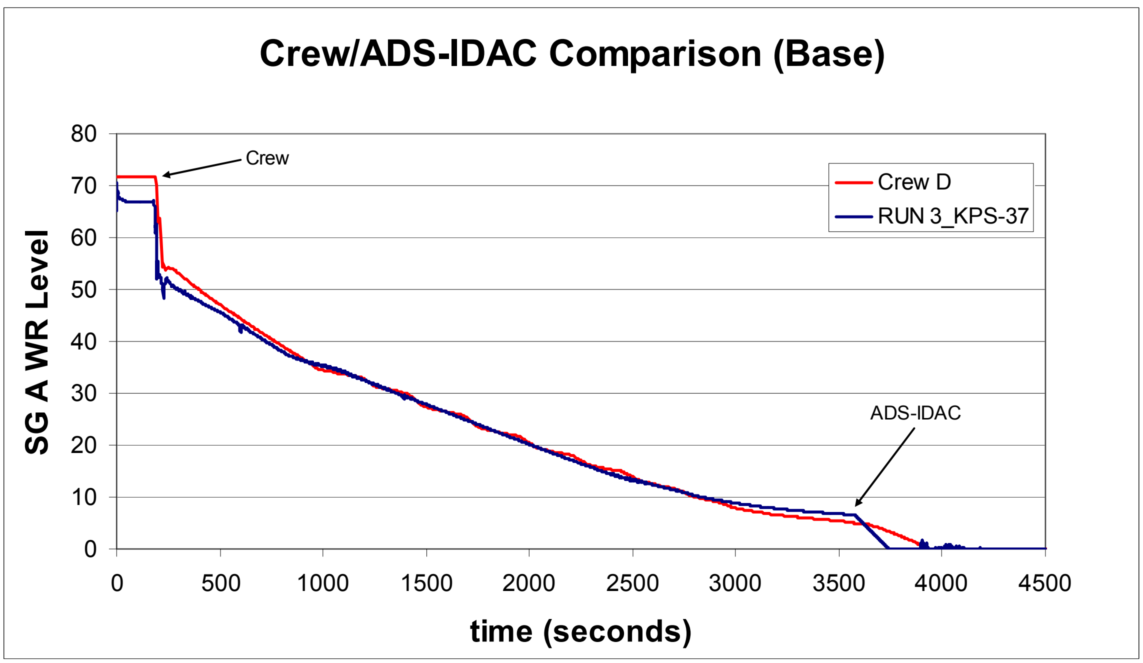


Figure 72 - Crew D (Base LOFW) SG A Wide Range Water Level

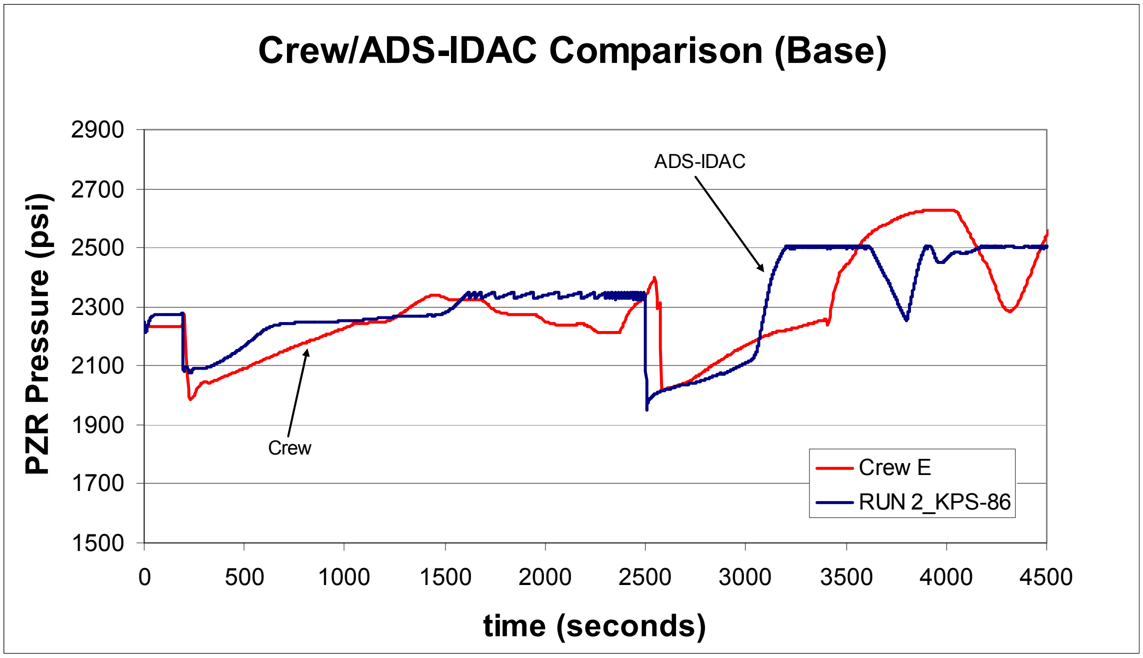


Figure 73 - Crew E (Base LOFW) Pressurizer Pressure

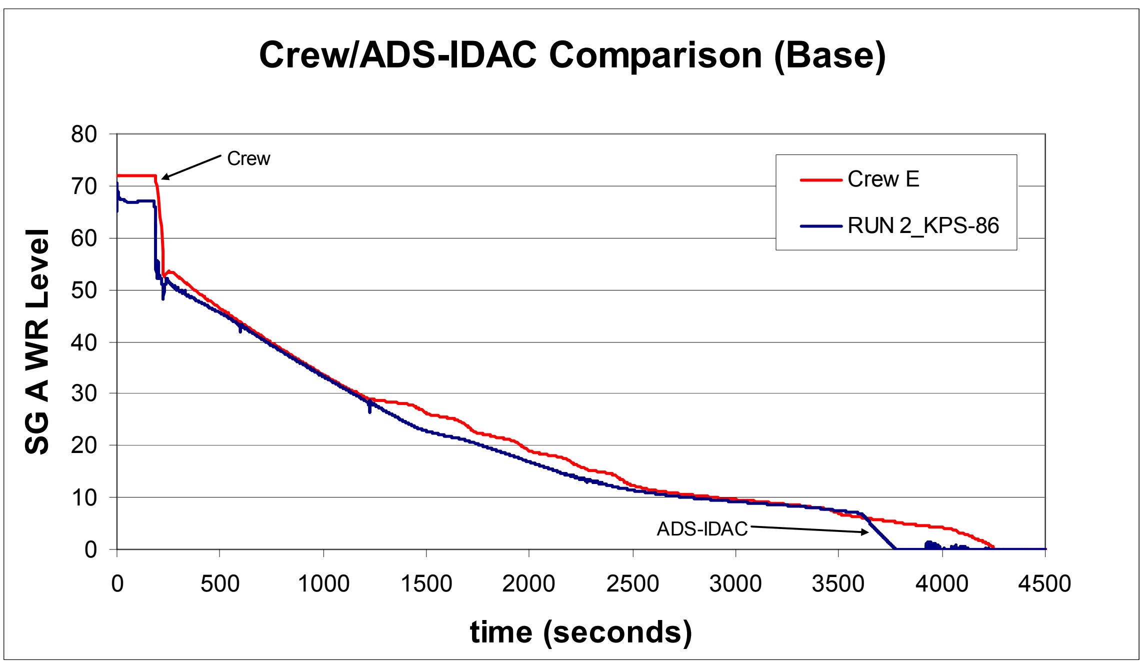


Figure 74 - Crew E (Base LOFW) SG A Wide Range Water Level

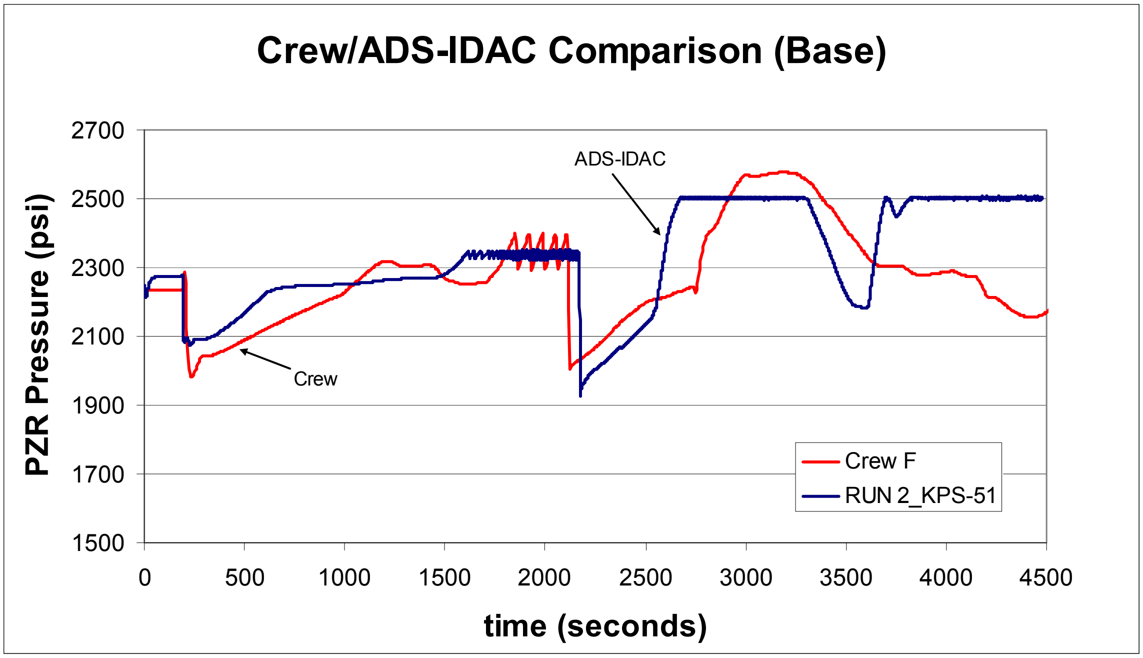


Figure 75 - Crew F (Base LOFW) Pressurizer Pressure

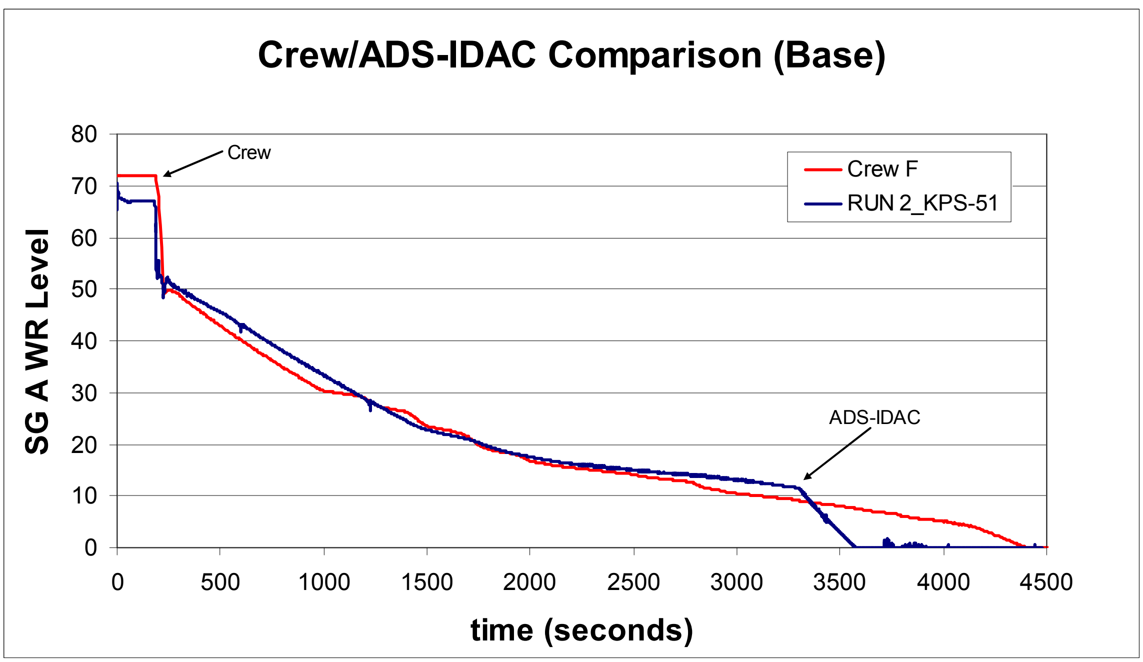


Figure 76 - Crew F (Base LOFW) SG A Wide Range Water Level

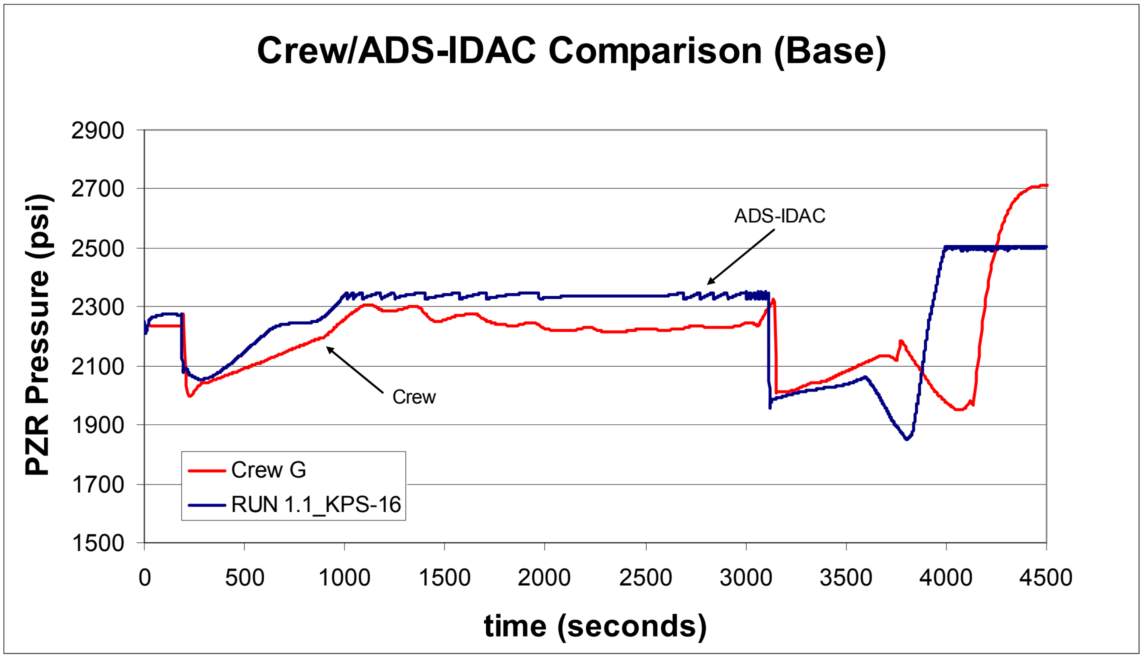


Figure 77 - Crew G (Base LOFW) Pressurizer Pressure

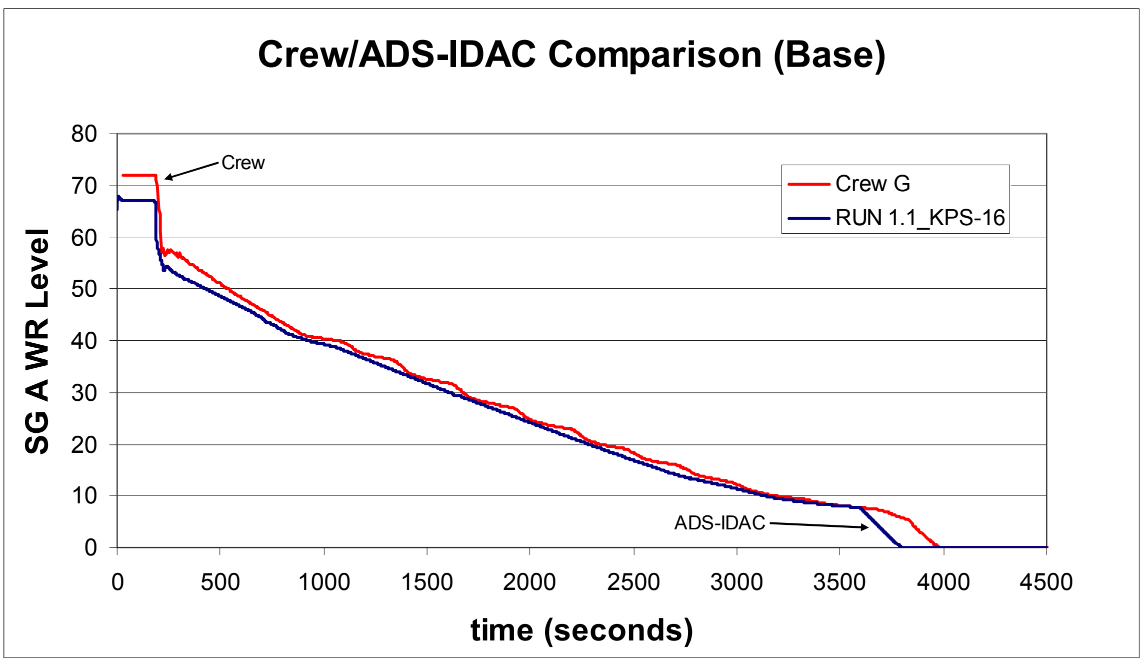


Figure 78 - Crew G (Base LOFW) SG A Wide Range Water Level

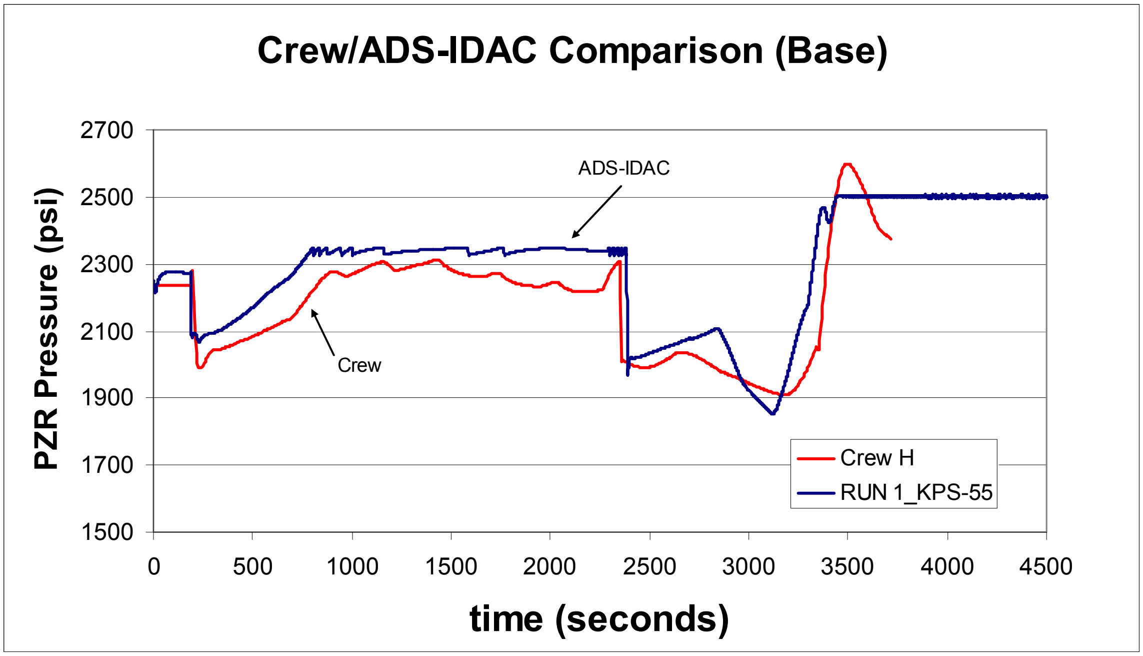


Figure 79 - Crew H (Base LOFW) Pressurizer Pressure

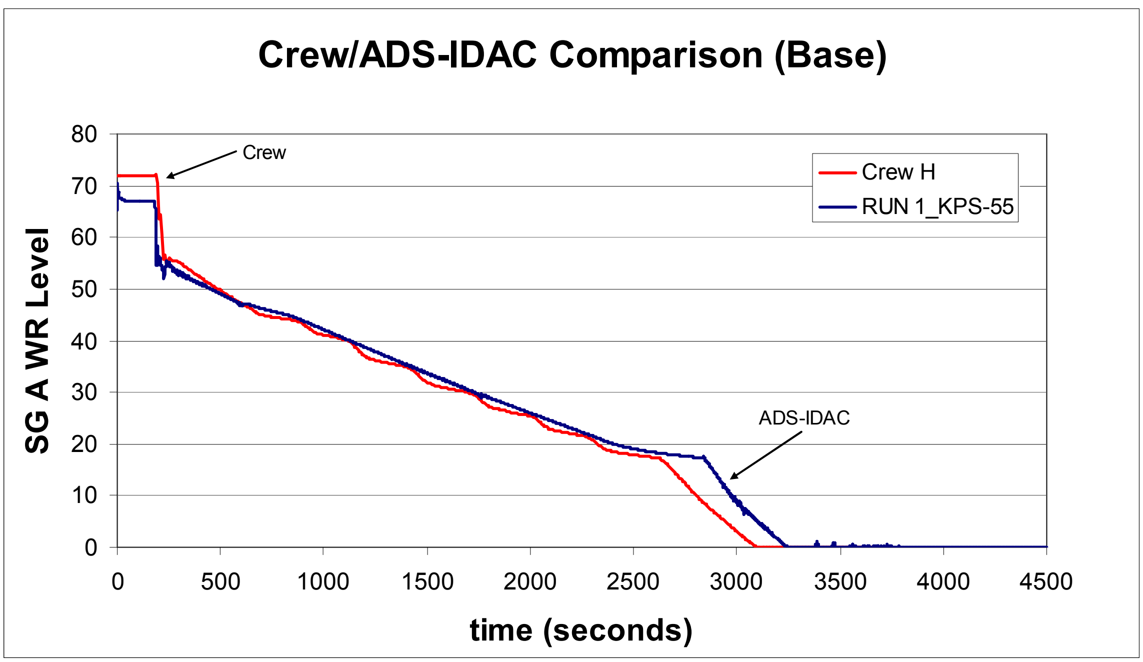


Figure 80 - Crew H (Base LOFW) SG A Wide Range Water Level

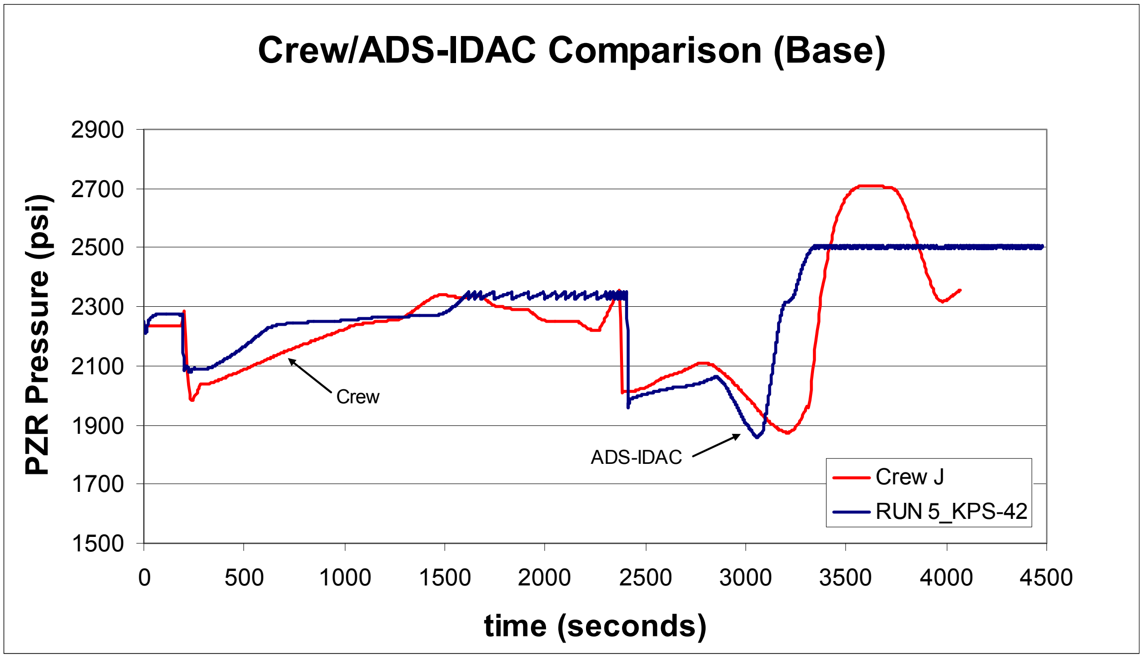


Figure 81 - Crew J (Base LOFW) Pressurizer Pressure

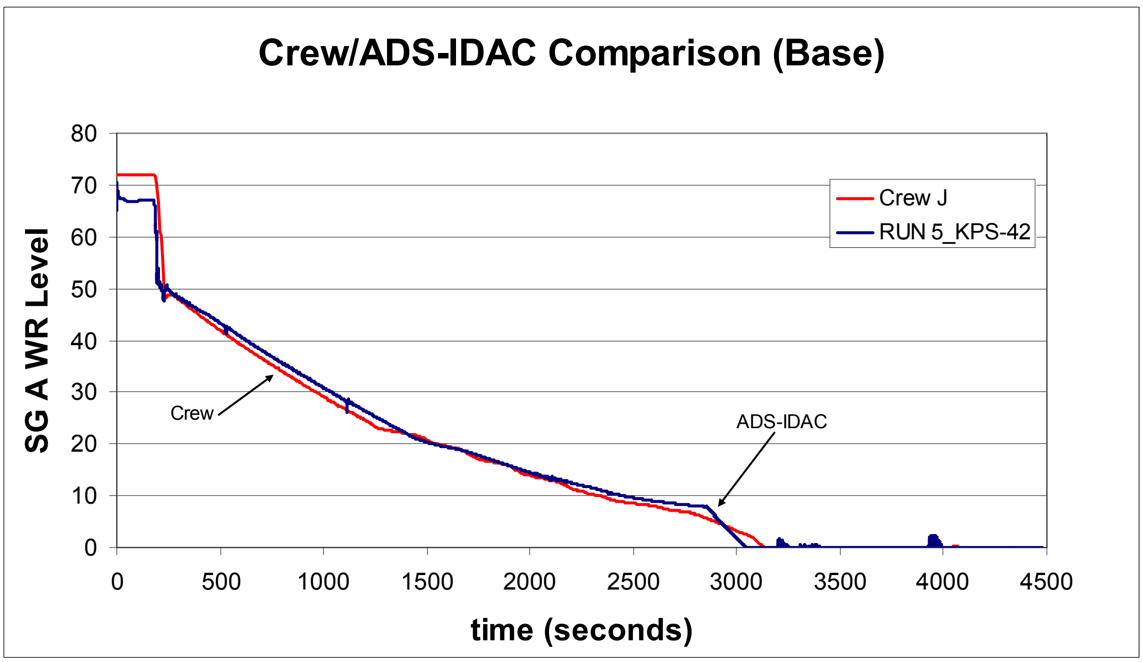


Figure 82 - Crew J (Base LOFW) SG A Wide Range Water Level

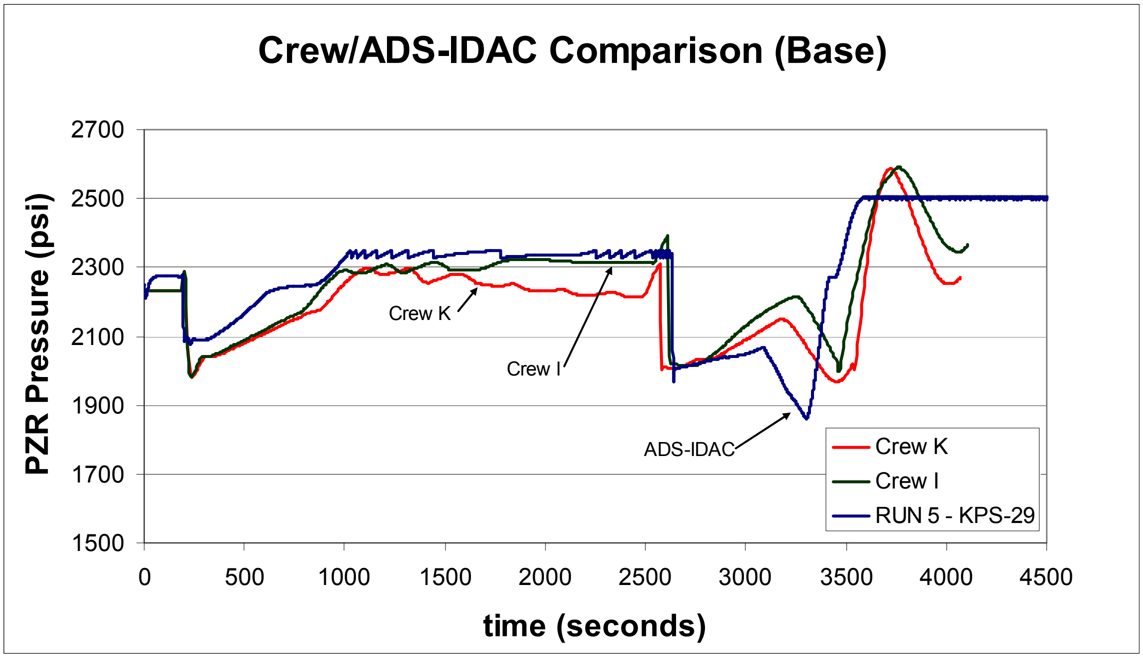


Figure 83 - Crews K & I (Base LOFW) Pressurizer Pressure

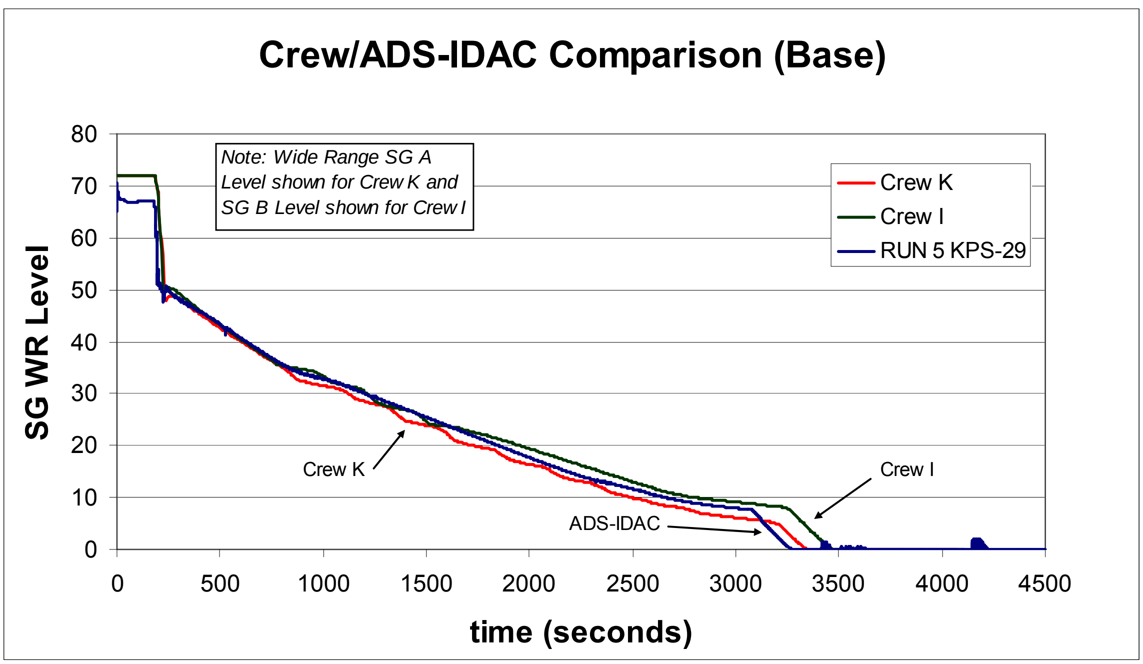


Figure 84 – Crews K & I (Base LOFW) SG A Wide Range Water Level

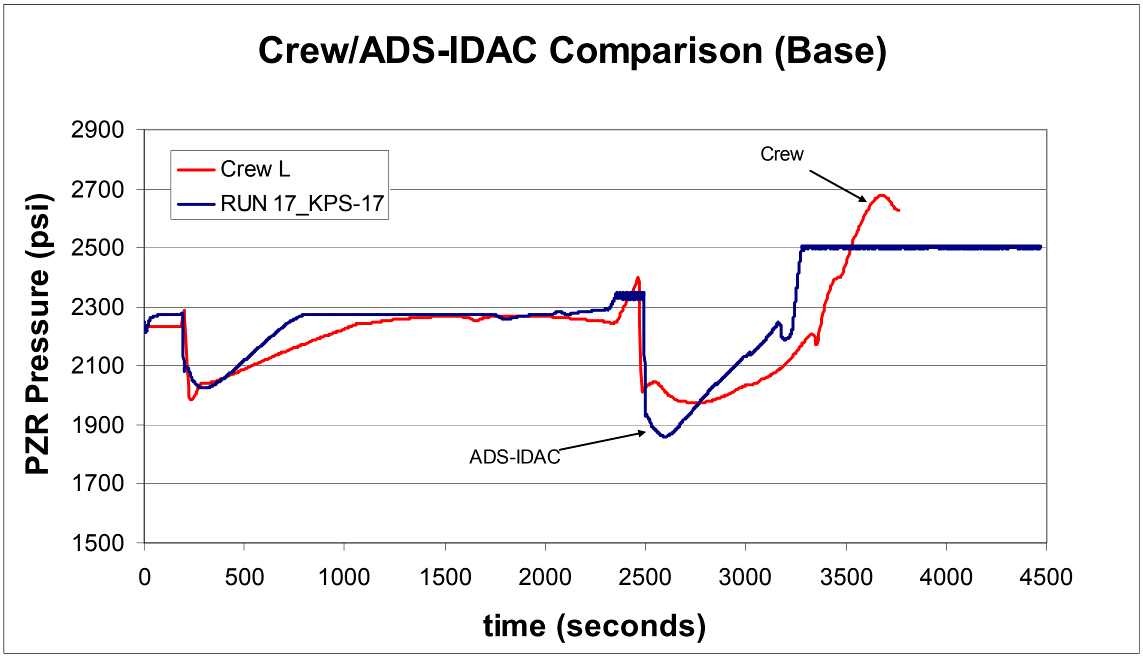


Figure 85 - Crew L (Base LOFW) Pressurizer Pressure

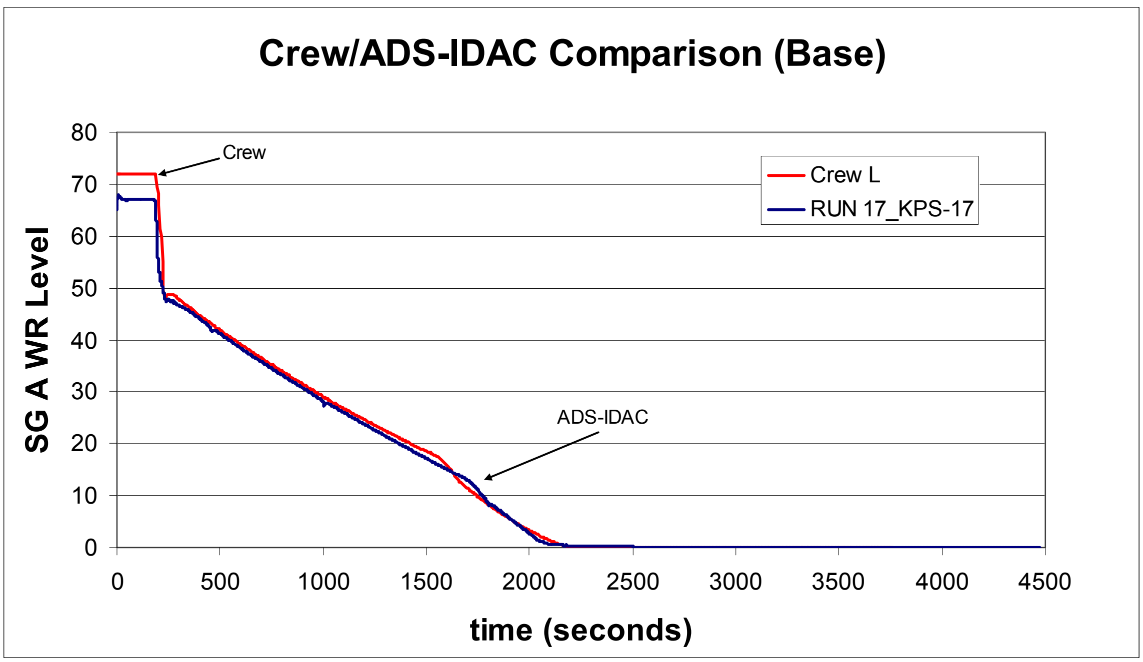


Figure 86 - Crew L (Base LOFW) SG A Wide Range Water Level

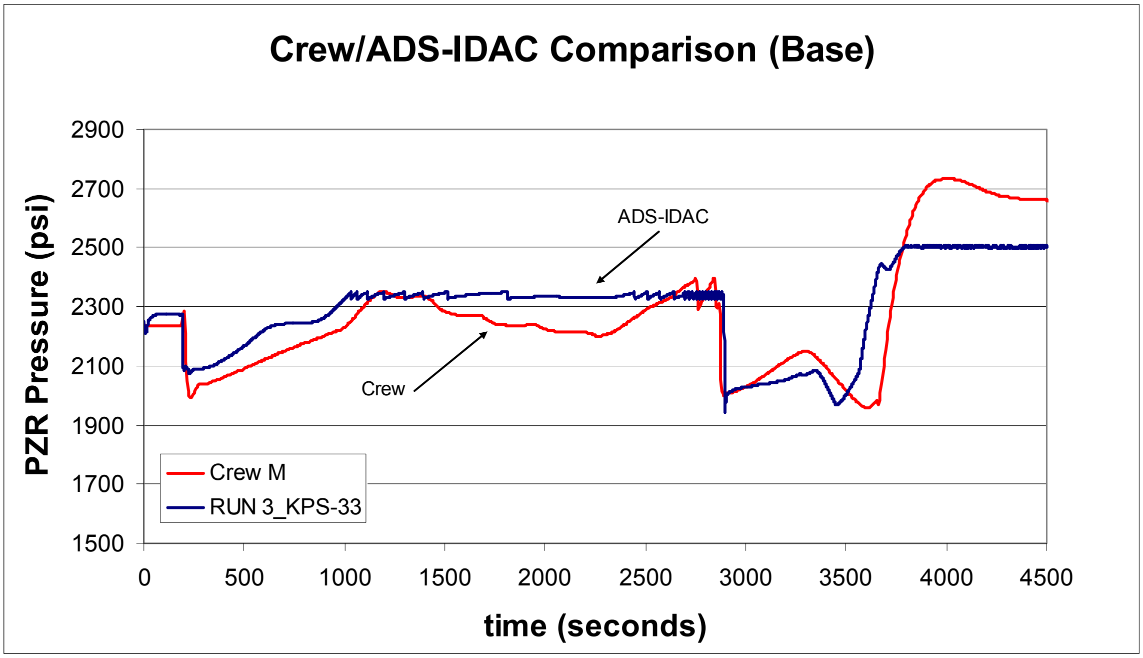


Figure 87 - Crew M (Base LOFW) Pressurizer Pressure

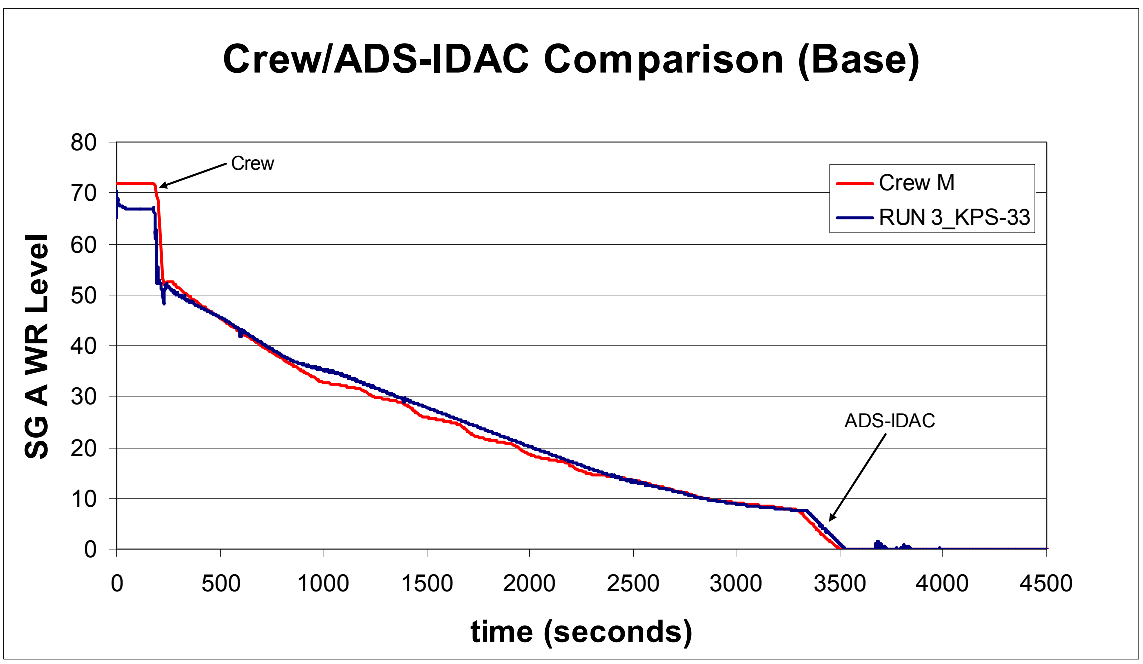


Figure 88 - Crew M (Base LOFW) SG A Wide Range Water Level

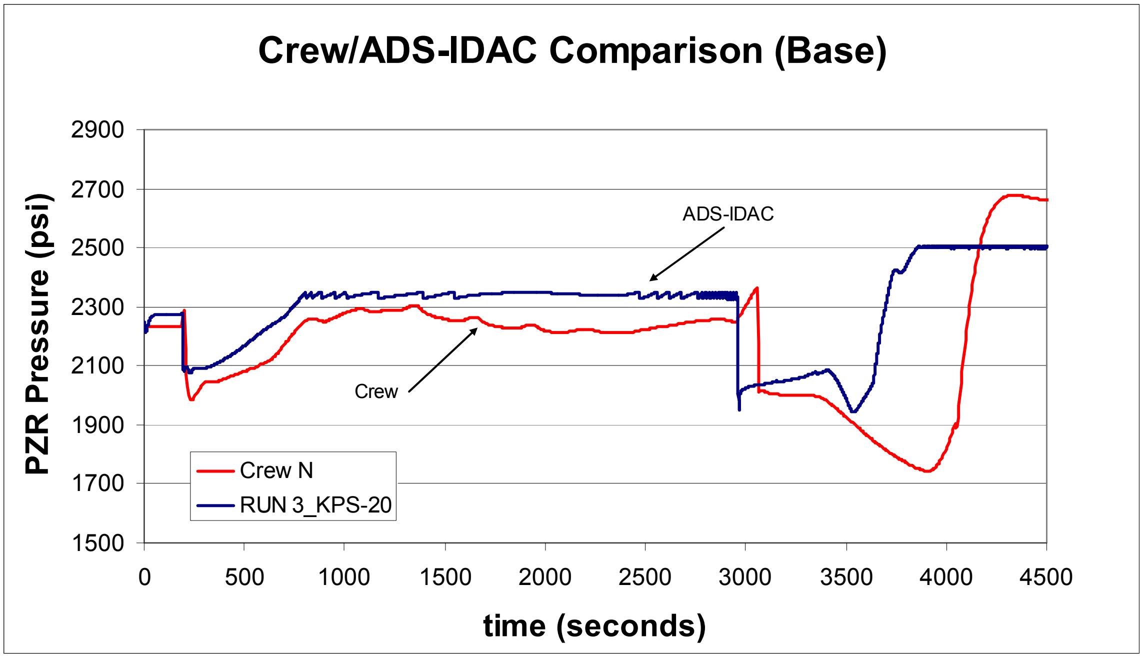


Figure 89 - Crew N (Base LOFW) Pressurizer Pressure

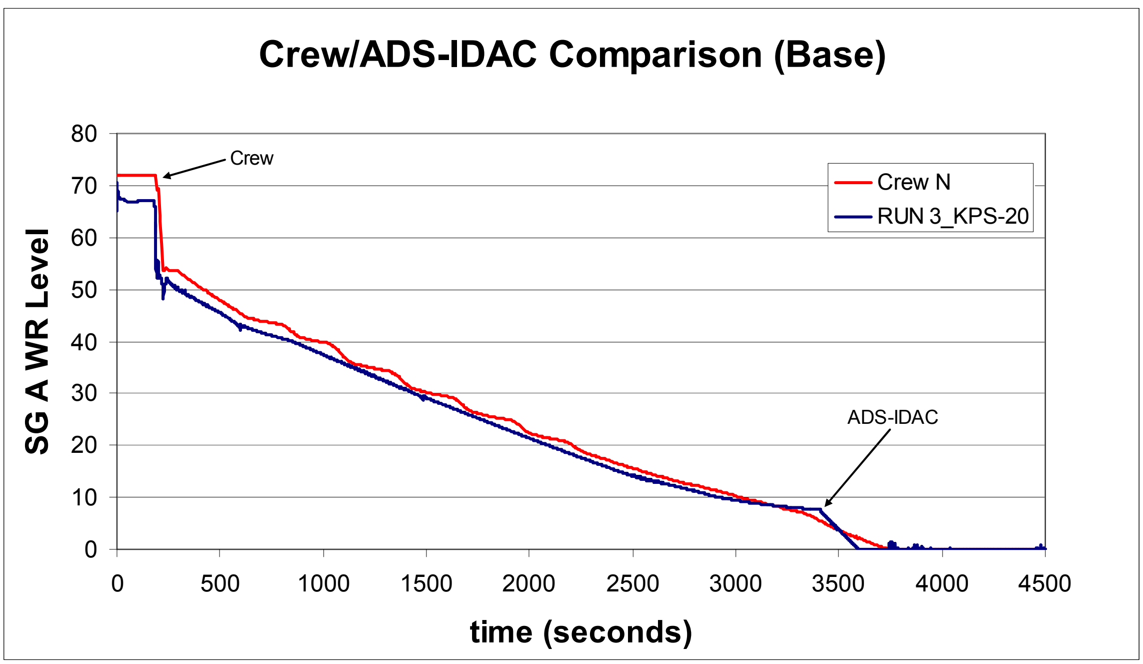


Figure 90 - Crew N (Base LOFW) SG A Wide Range Water Level

Appendix J – LOFW Complex Case Comparison

Appendix J provides a series of graphs that compare the performance of control room crews at the Halden FRESH simulator facility during the complex case loss of feedwater event to the results of the re-calibrated ADS-IDAC model. Although the ADS-IDAC results were obtained after the loss of feedwater empirical study data was reviewed and the ADS-IDAC knowledge base adjusted, this comparison demonstrates that the model is capable of representing a wide range of crew-to-crew variabilities with a relatively small number of branching events.

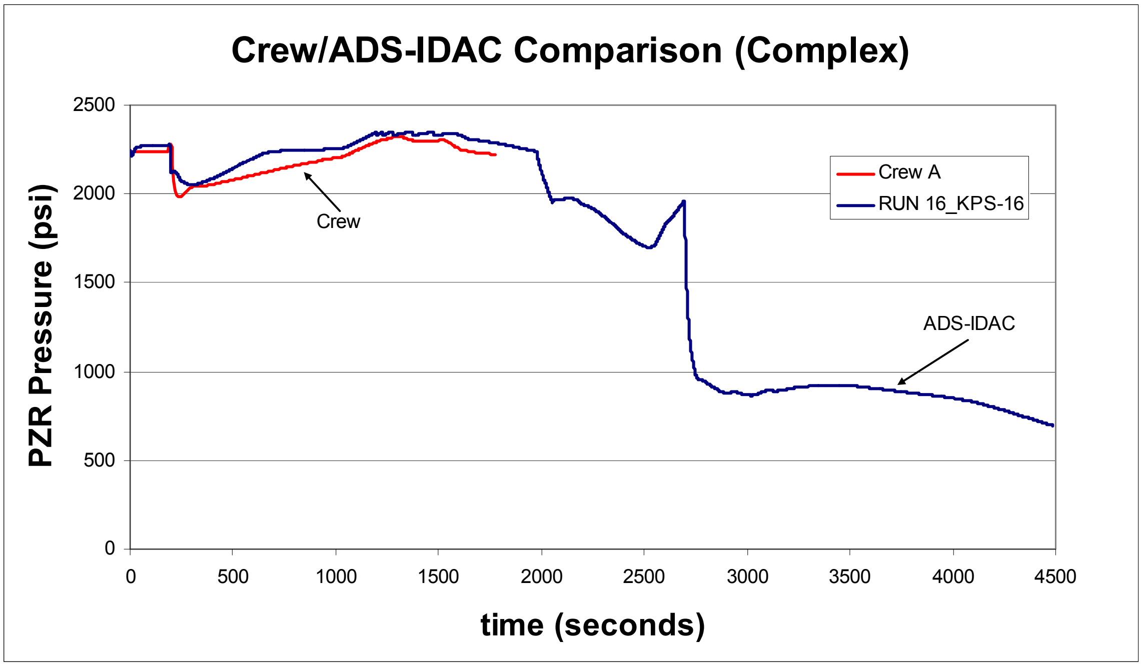


Figure 91 - Crew A (Complex LOFW) Pressurizer Pressure

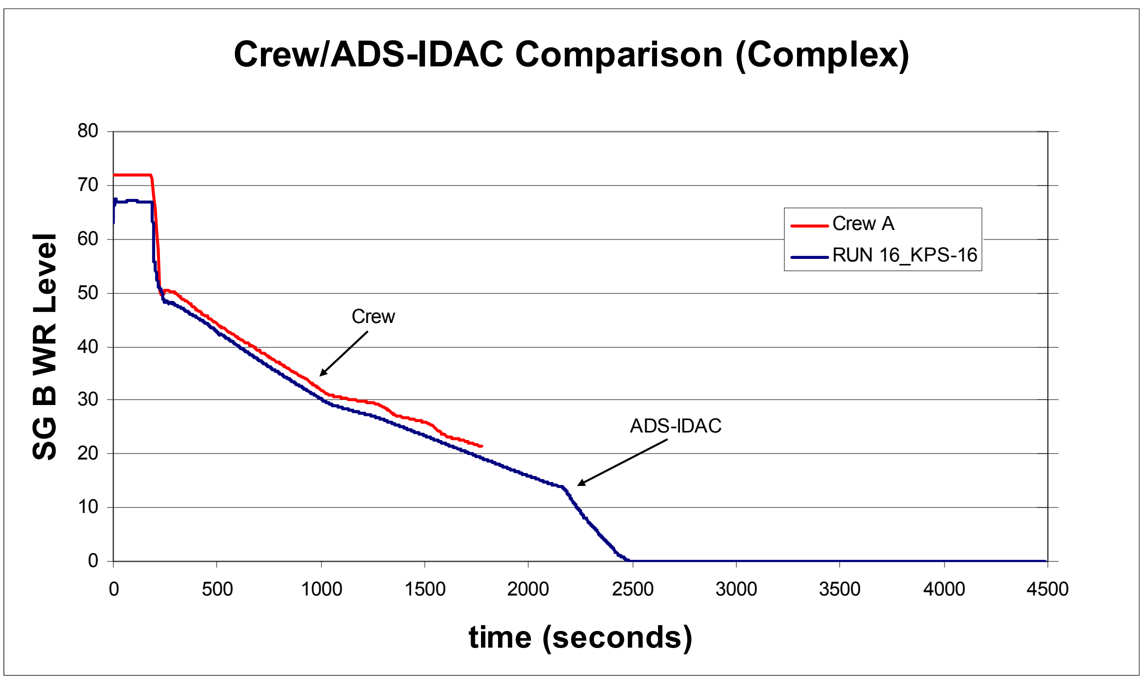


Figure 92 - Crew A (Complex LOFW) SG B Wide Range Water Level

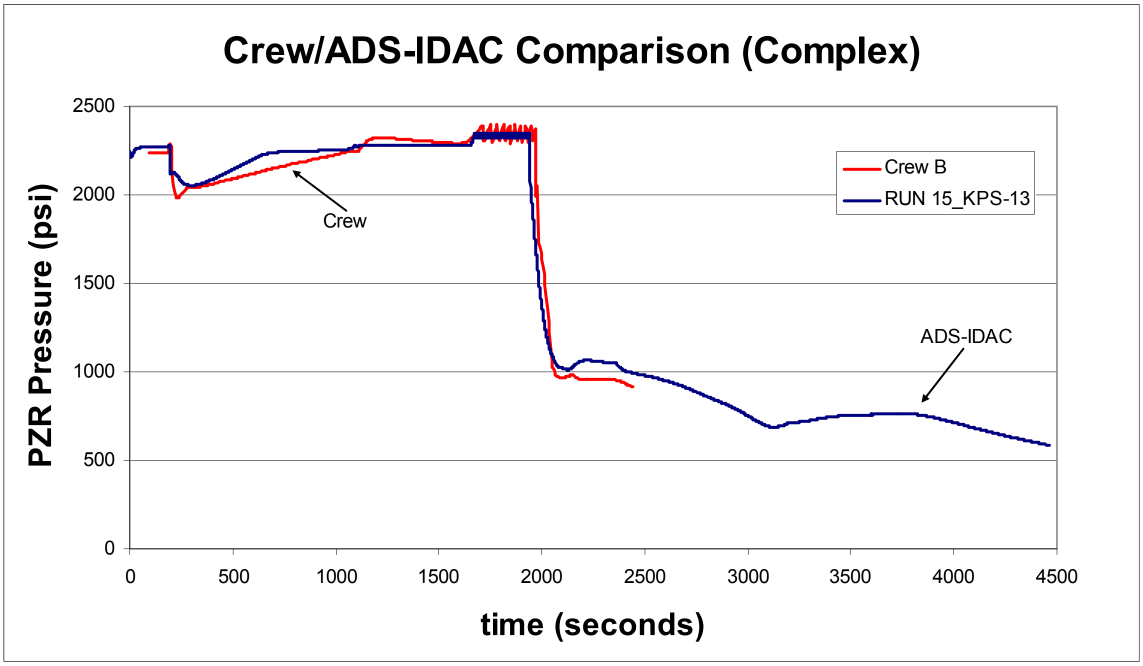


Figure 93 - Crew B (Complex LOFW) Pressurizer Pressure

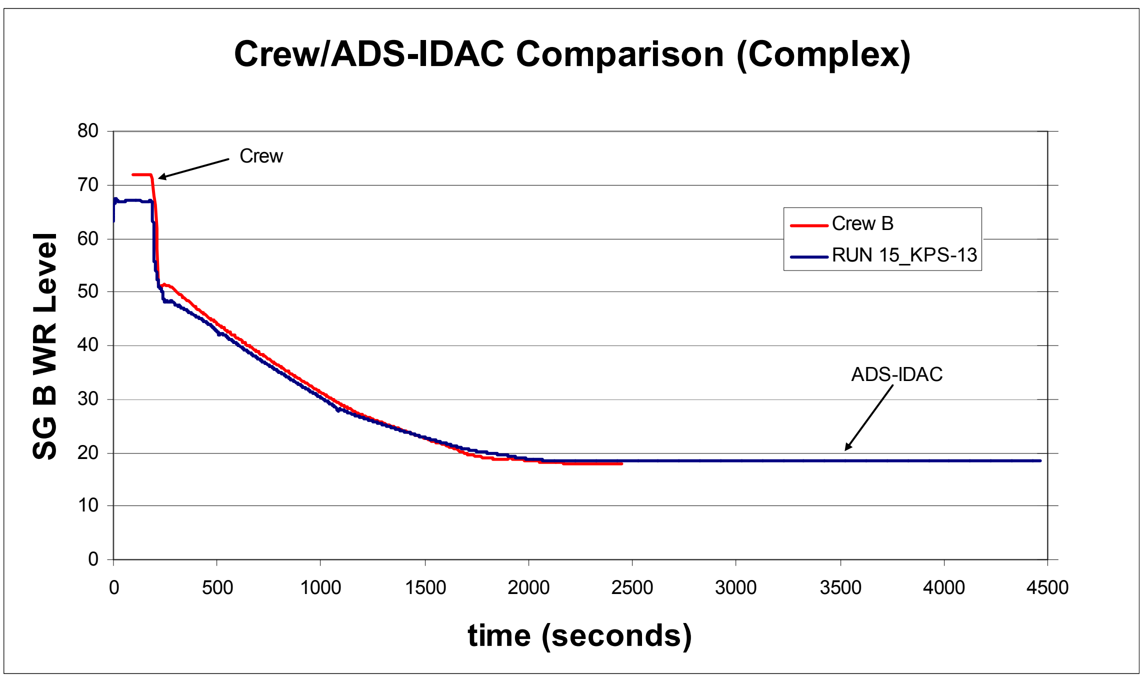


Figure 94 - Crew B (Complex LOFW) SG B Wide Range Water Level

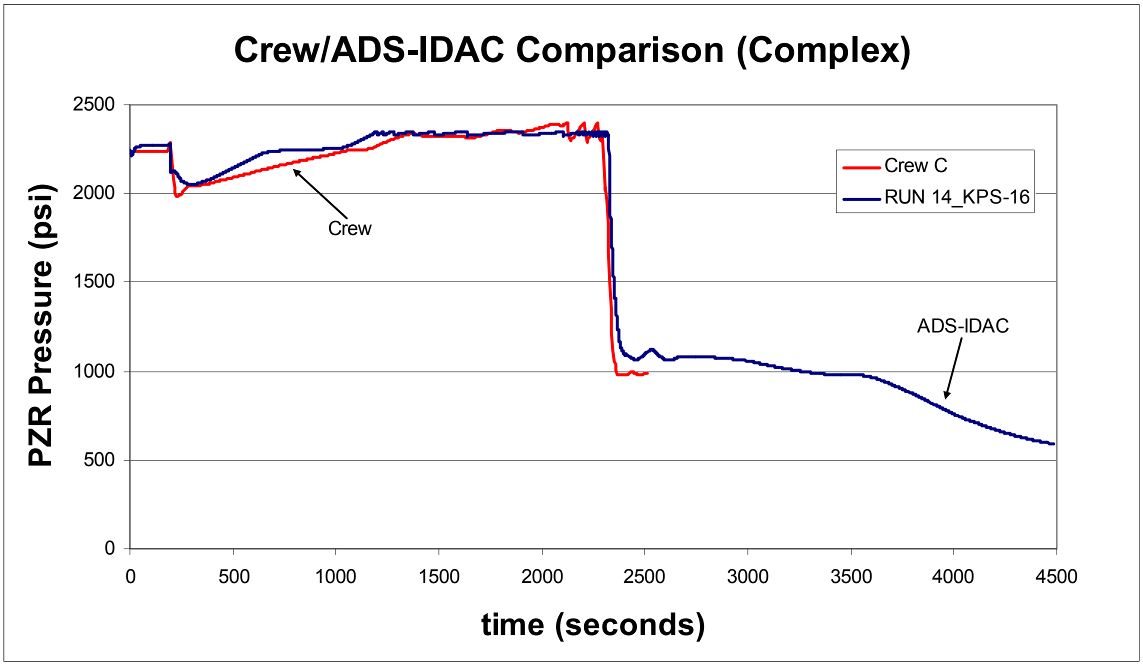


Figure 95 - Crew C (Complex LOFW) Pressurizer Pressure

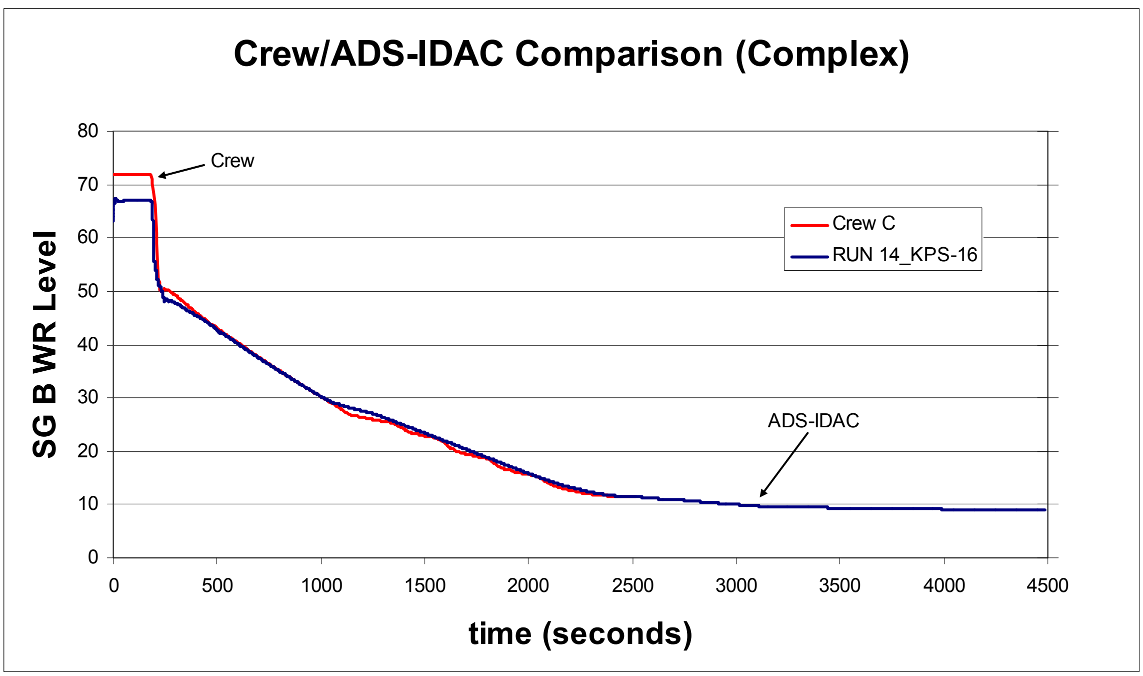


Figure 96 - Crew C (Complex LOFW) SG B Wide Range Water Level

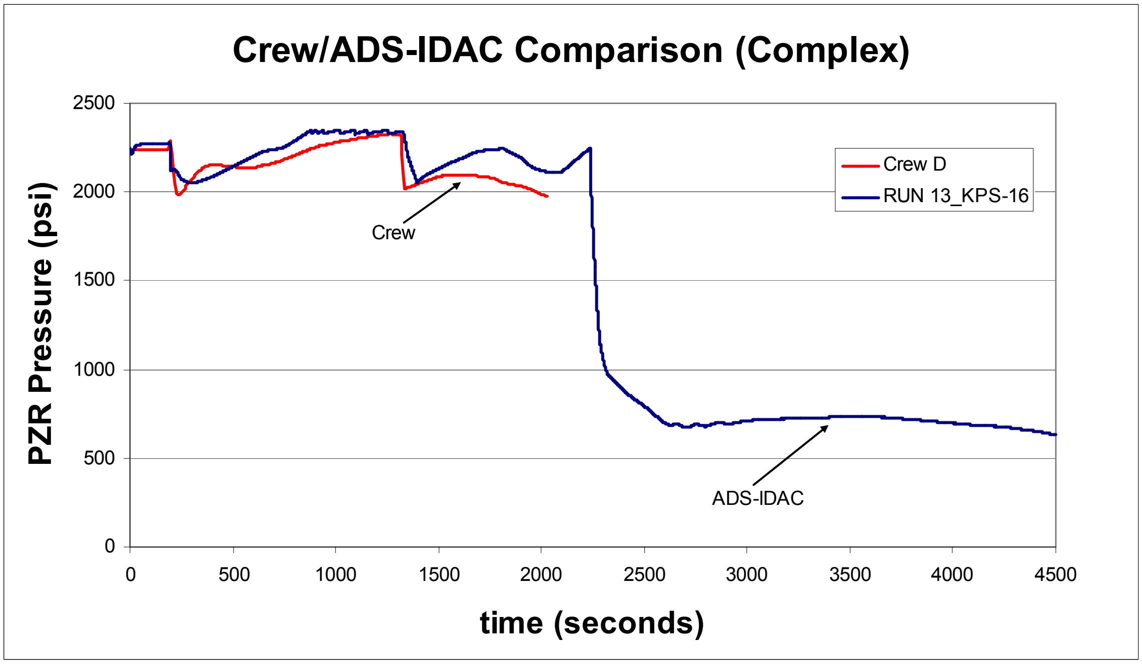


Figure 97 - Crew D (Complex LOFW) Pressurizer Pressure

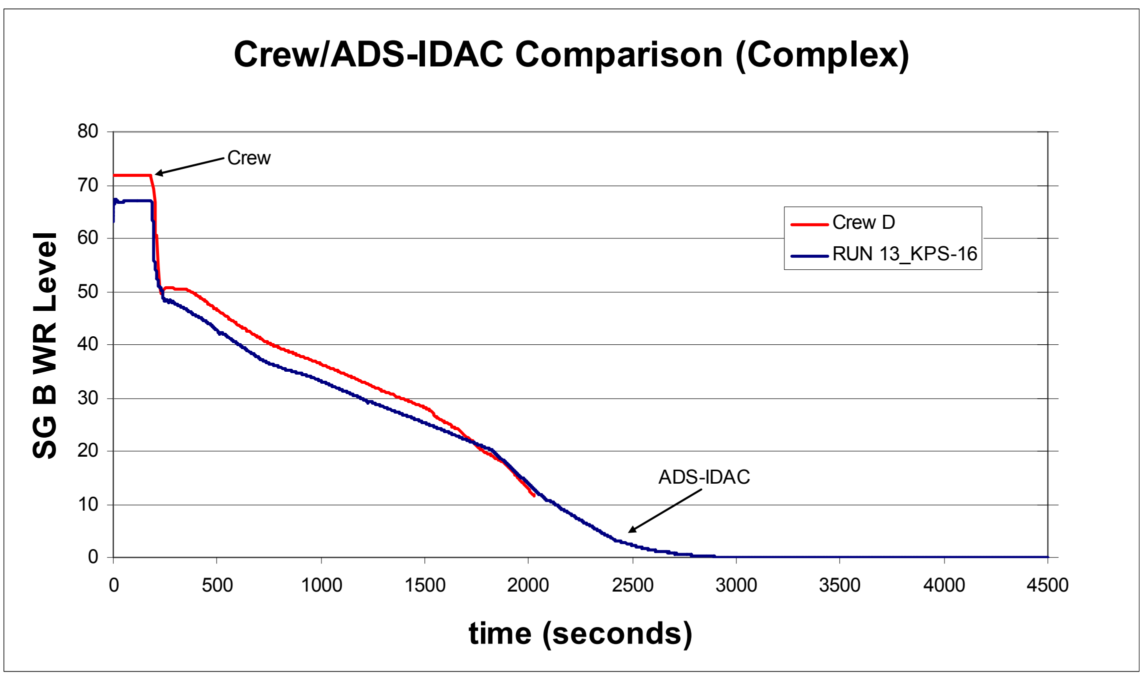


Figure 98 - Crew D (Complex LOFW) SG B Wide Range Water Level

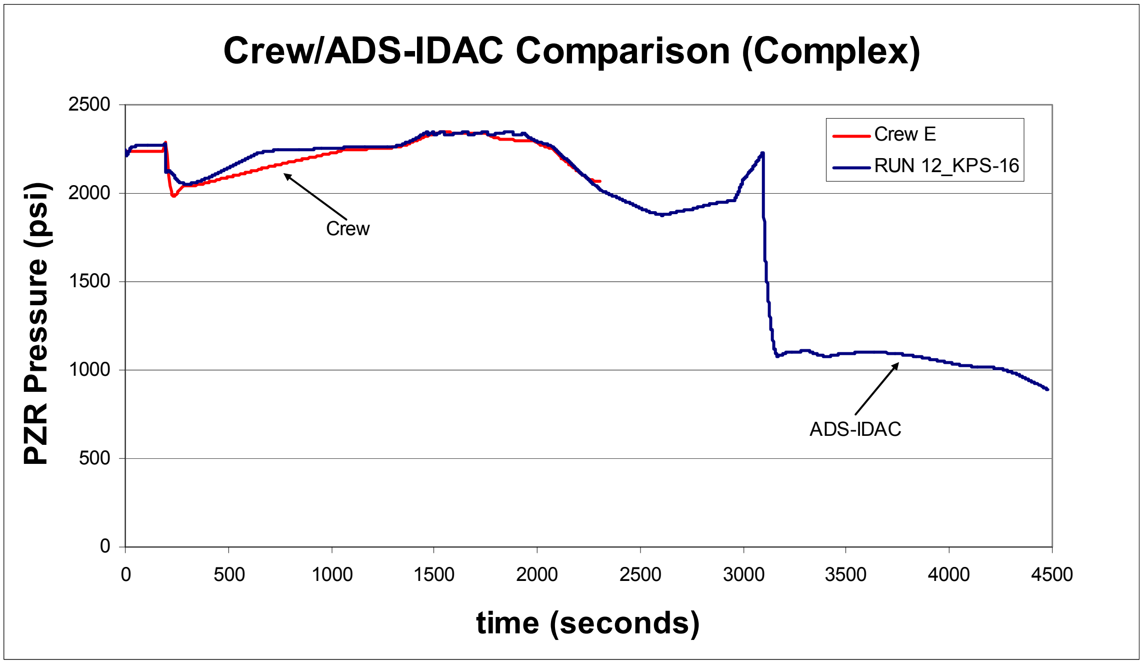


Figure 99 - Crew E (Complex LOFW) Pressurizer Pressure

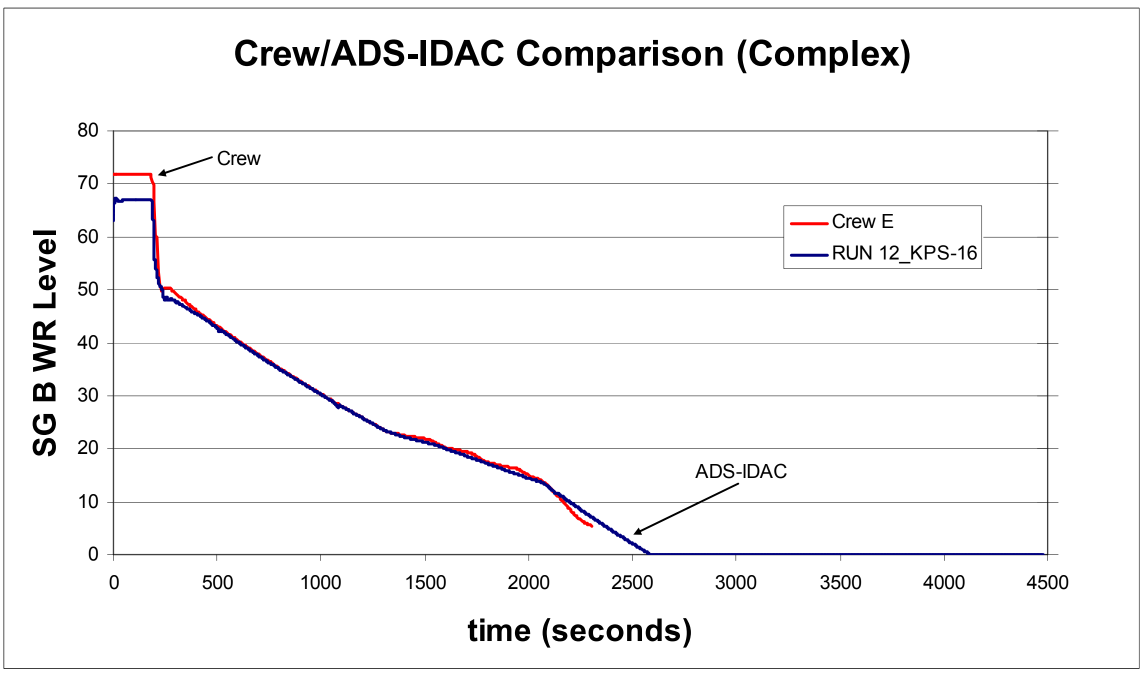


Figure 100 - Crew E (Complex LOFW) SG B Wide Range Water Level

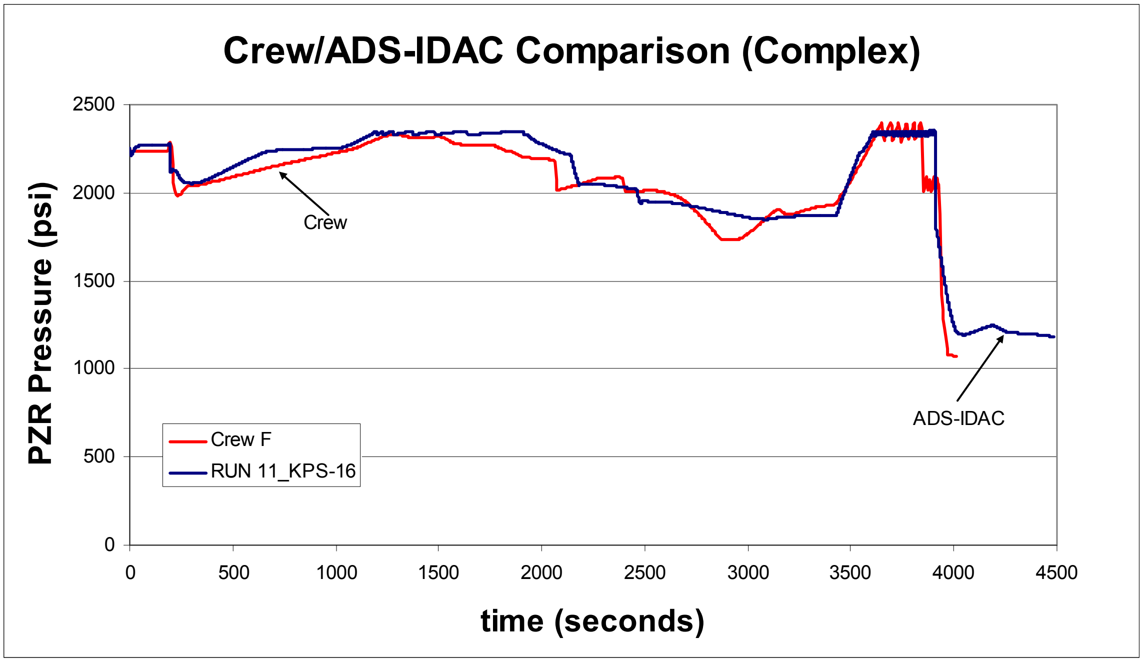


Figure 101 - Crew F (Complex LOFW) Pressurizer Pressure

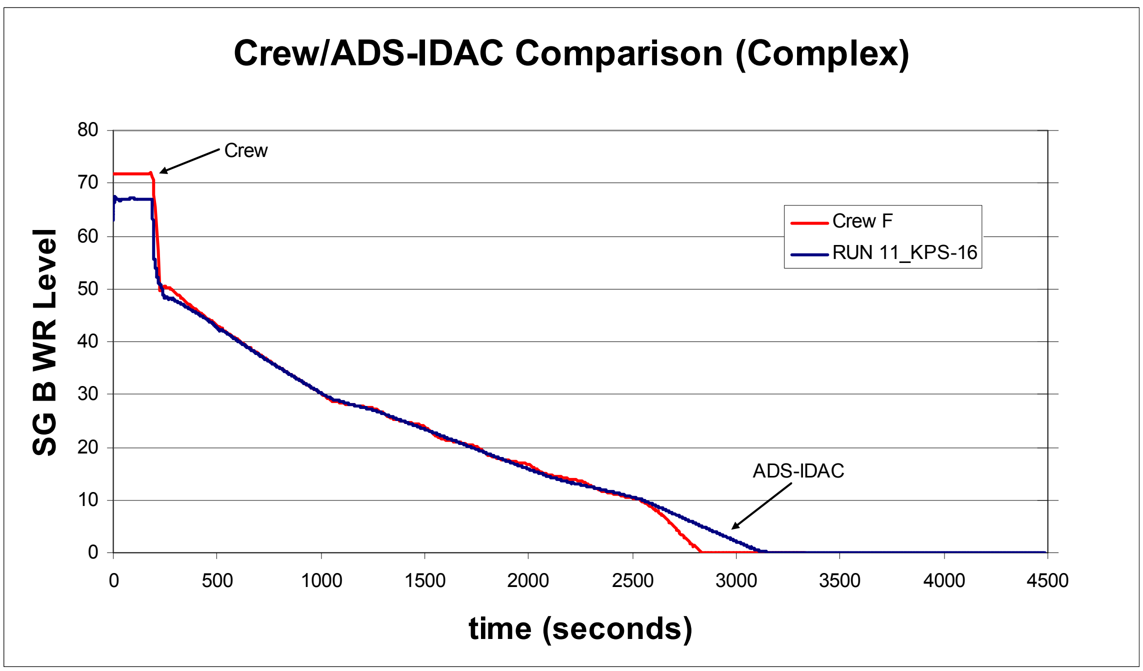


Figure 102 - Crew F (Complex LOFW) SG B Wide Range Water Level

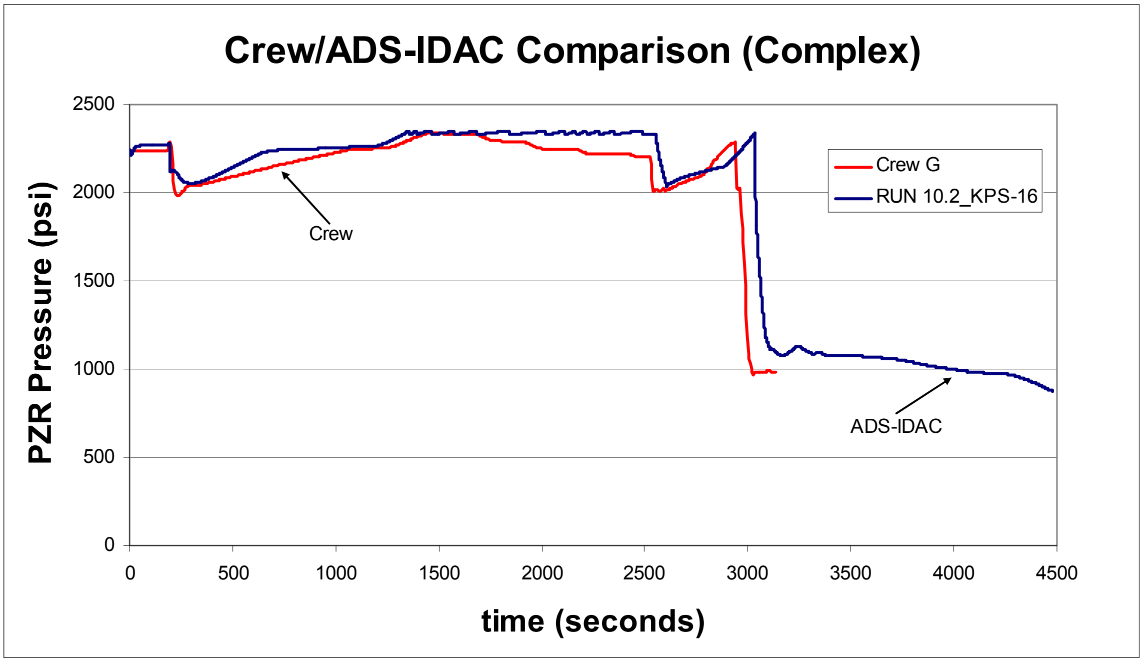


Figure 103 - Crew G (Complex LOFW) Pressurizer Pressure

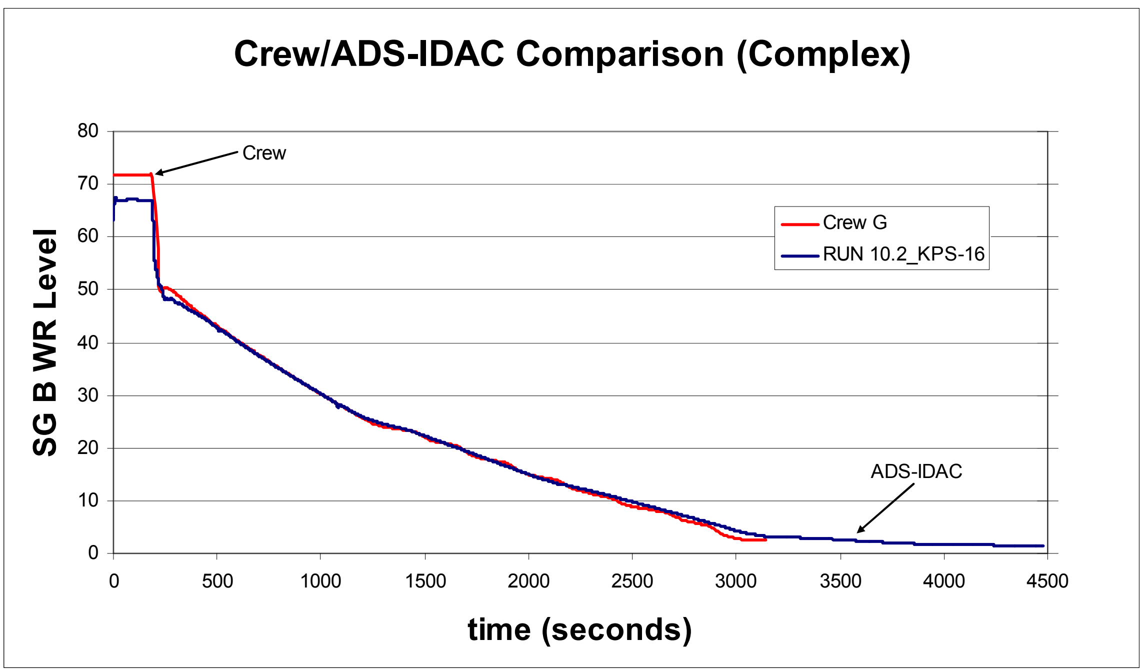


Figure 104 - Crew G (Complex LOFW) SG B Wide Range Water Level

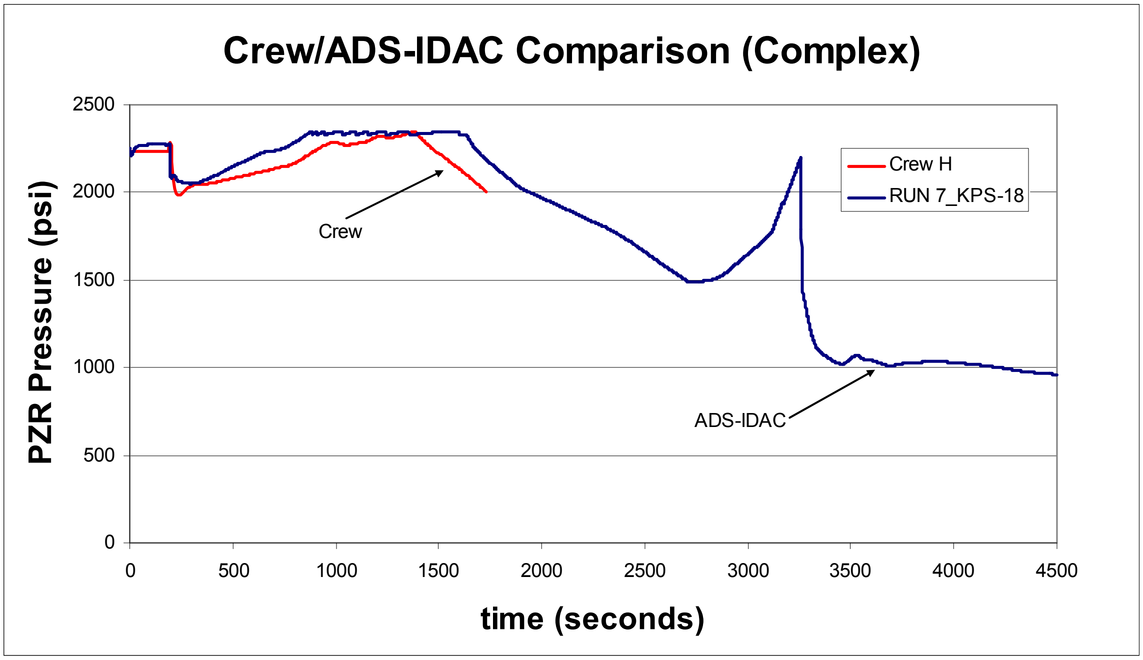


Figure 105 - Crew H (Complex LOFW) Pressurizer Pressure

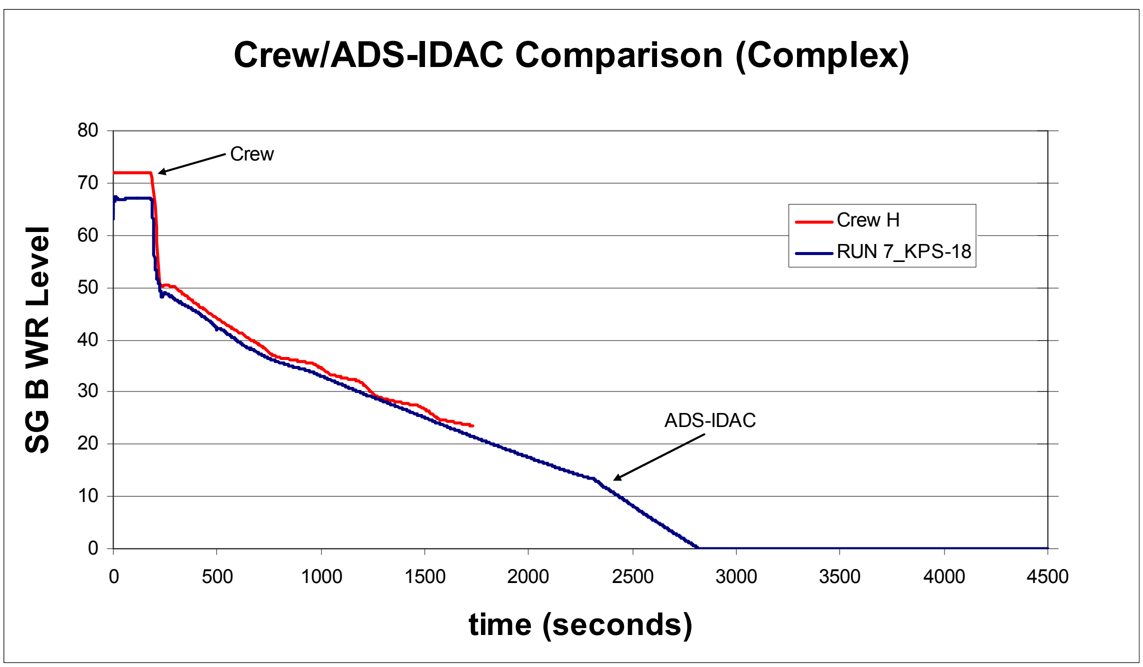


Figure 106 - Crew H (Complex LOFW) SG B Wide Range Water Level

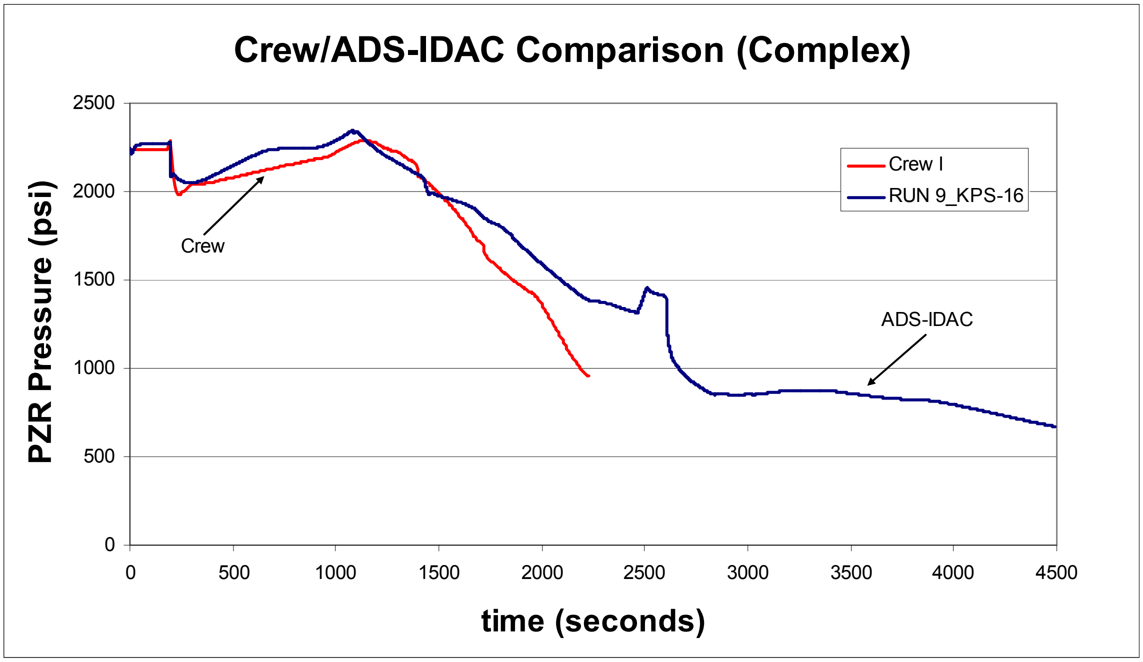


Figure 107 - Crew I (Complex LOFW) Pressurizer Pressure

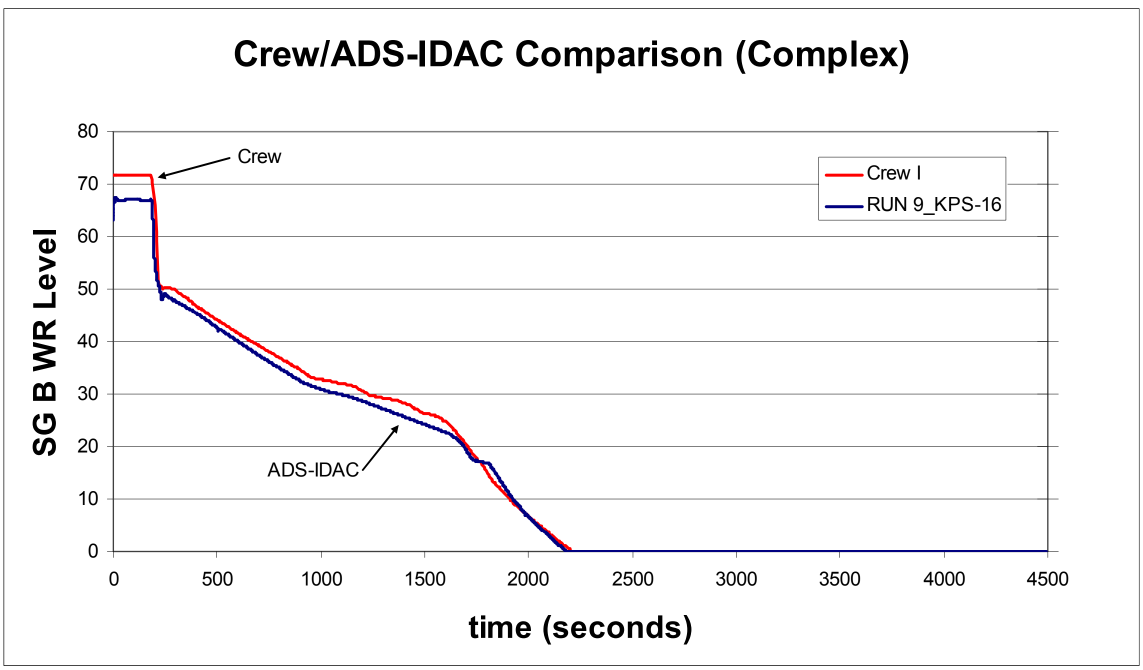


Figure 108 - Crew I (Complex LOFW) SG B Wide Range Water Level

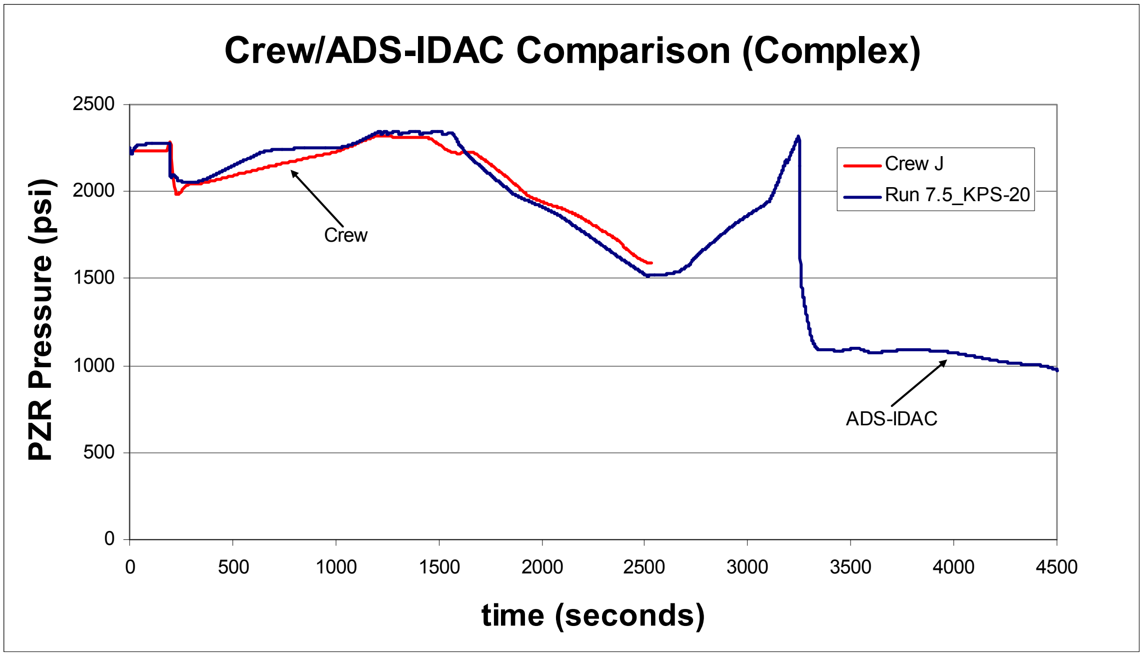


Figure 109 - Crew J (Complex LOFW) Pressurizer Pressure

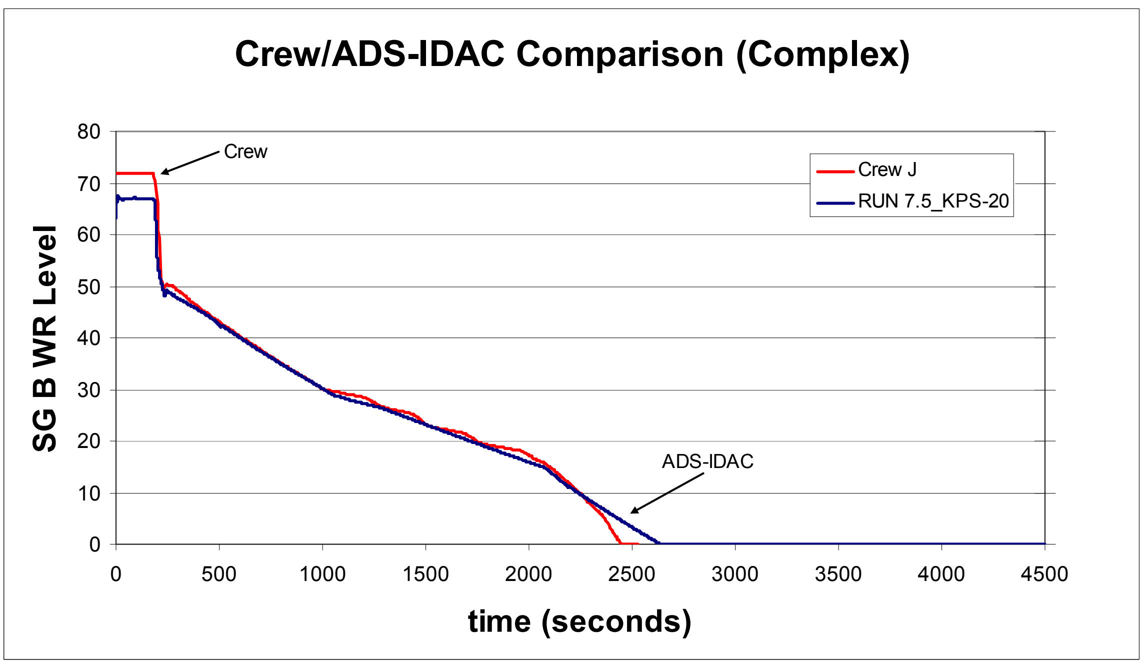


Figure 110 - Crew J (Complex LOFW) SG B Wide Range Water Level

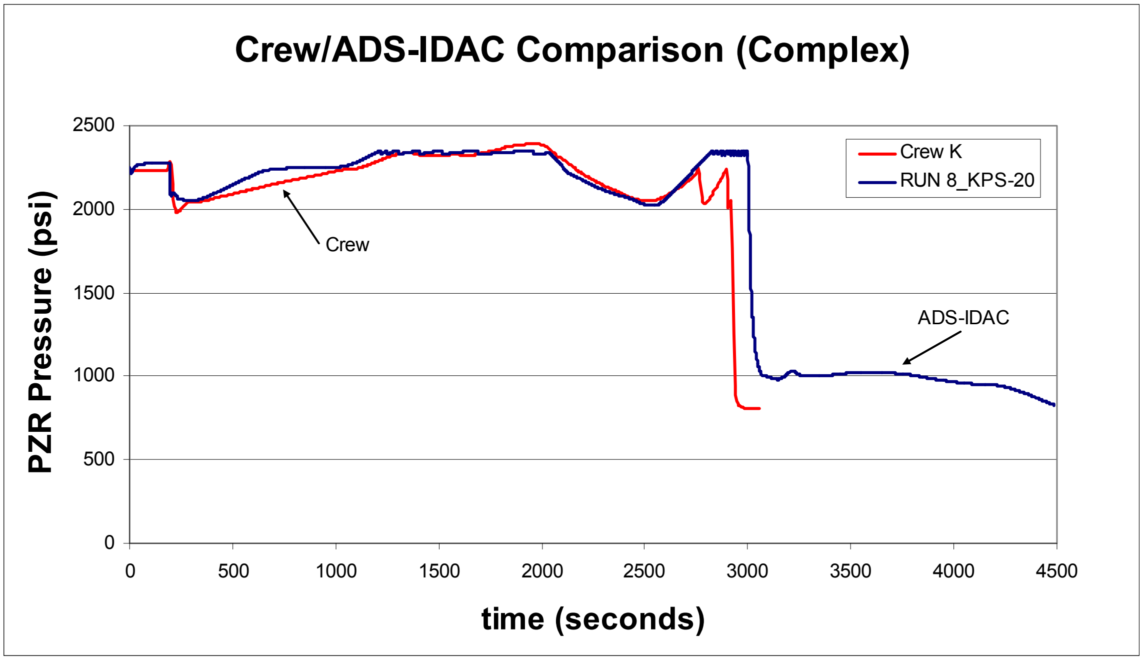


Figure 111 - Crew K (Complex LOFW) Pressurizer Pressure

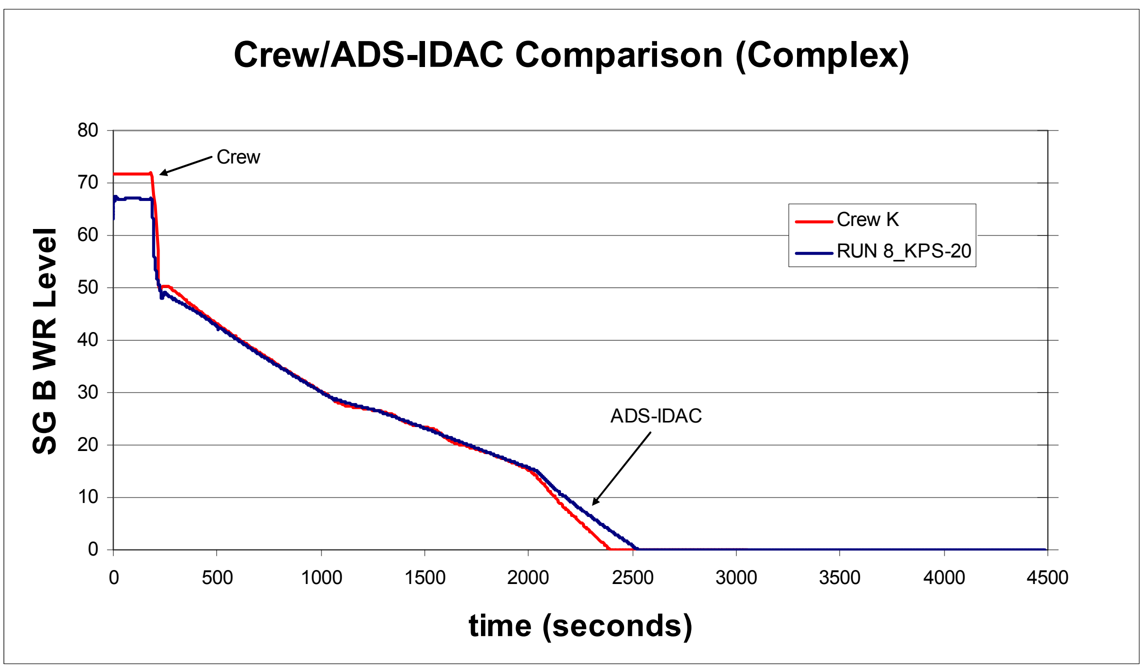


Figure 112 - Crew K (Complex LOFW) SG B Wide Range Water Level

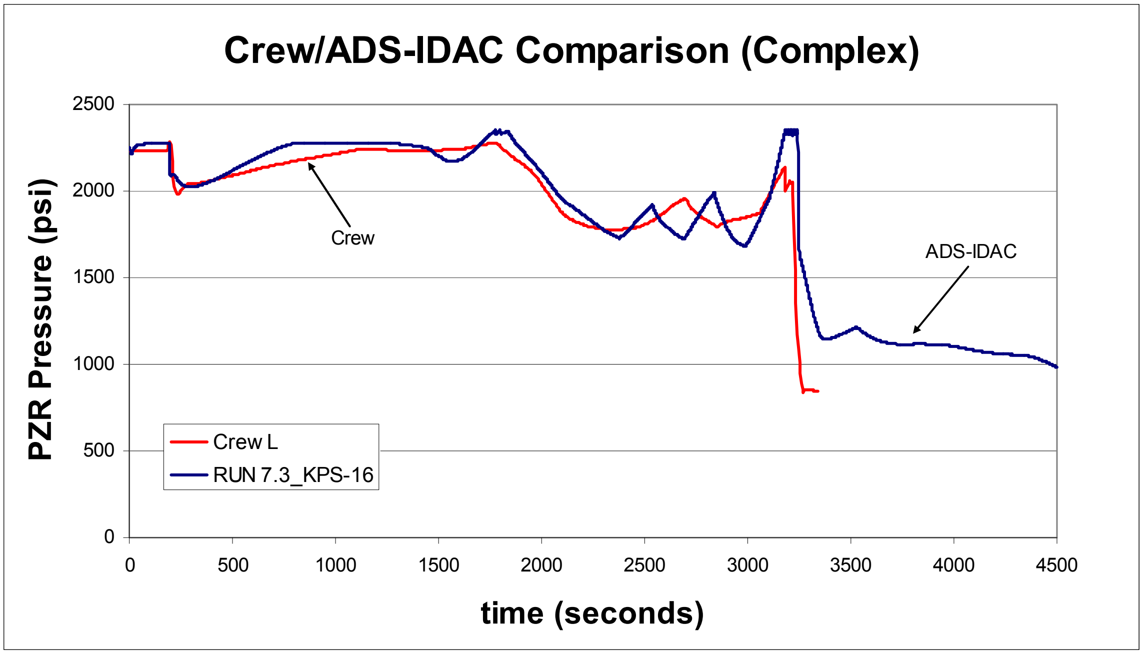


Figure 113 - Crew L (Complex LOFW) Pressurizer Pressure

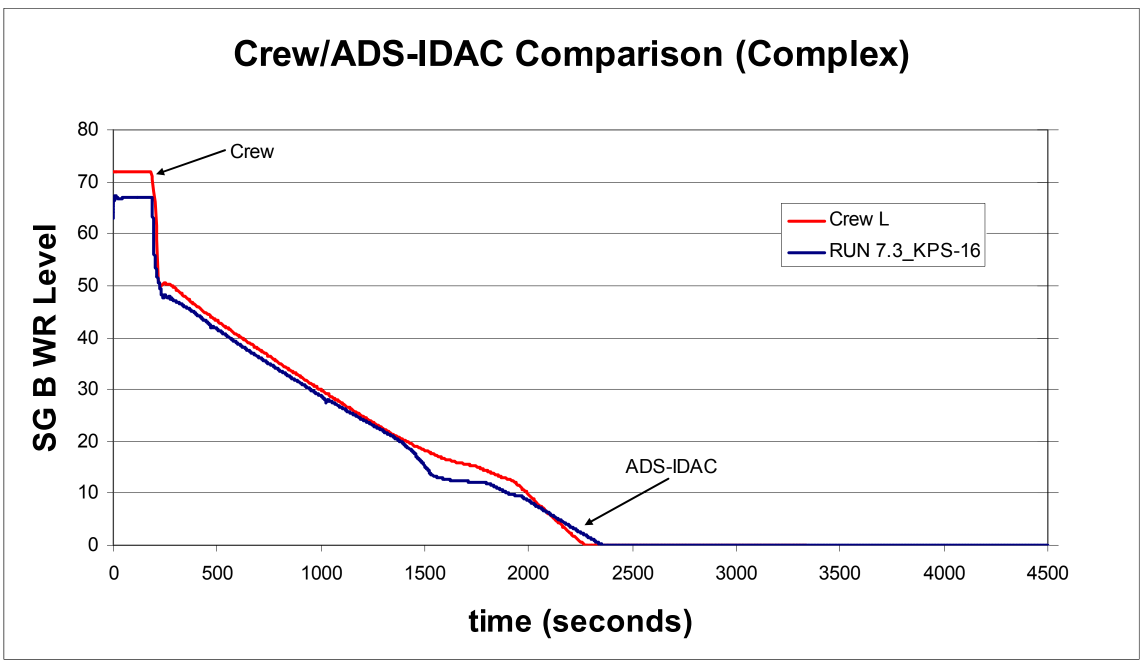


Figure 114 - Crew L (Complex LOFW) SG B Wide Range Water Level

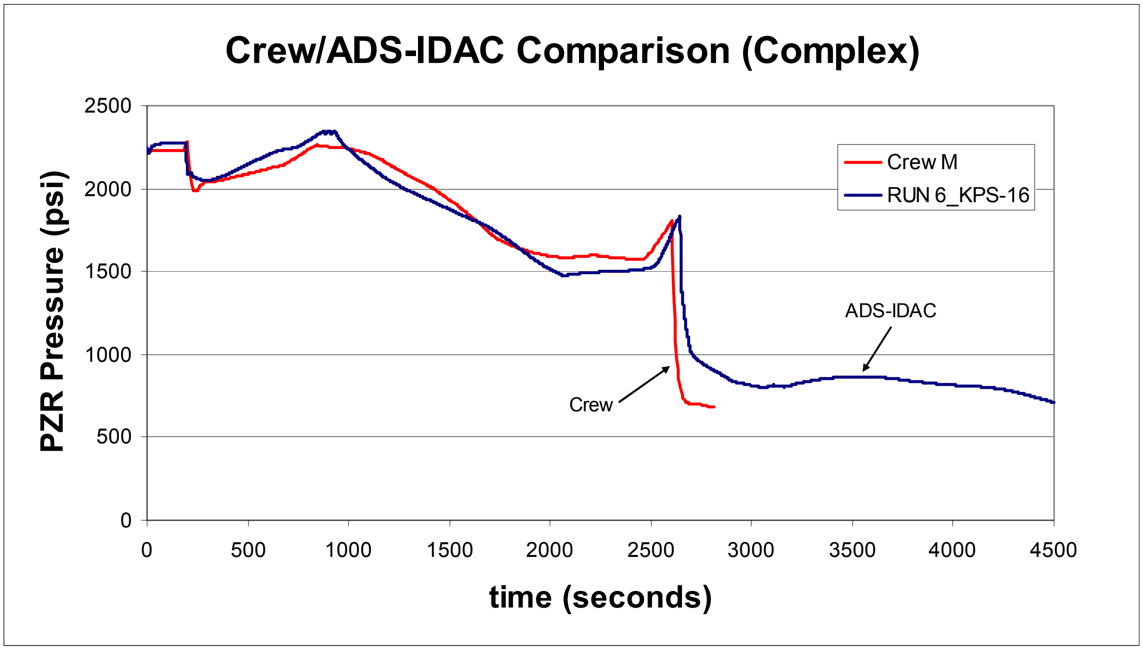


Figure 115 - Crew M (Complex LOFW) Pressurizer Pressure

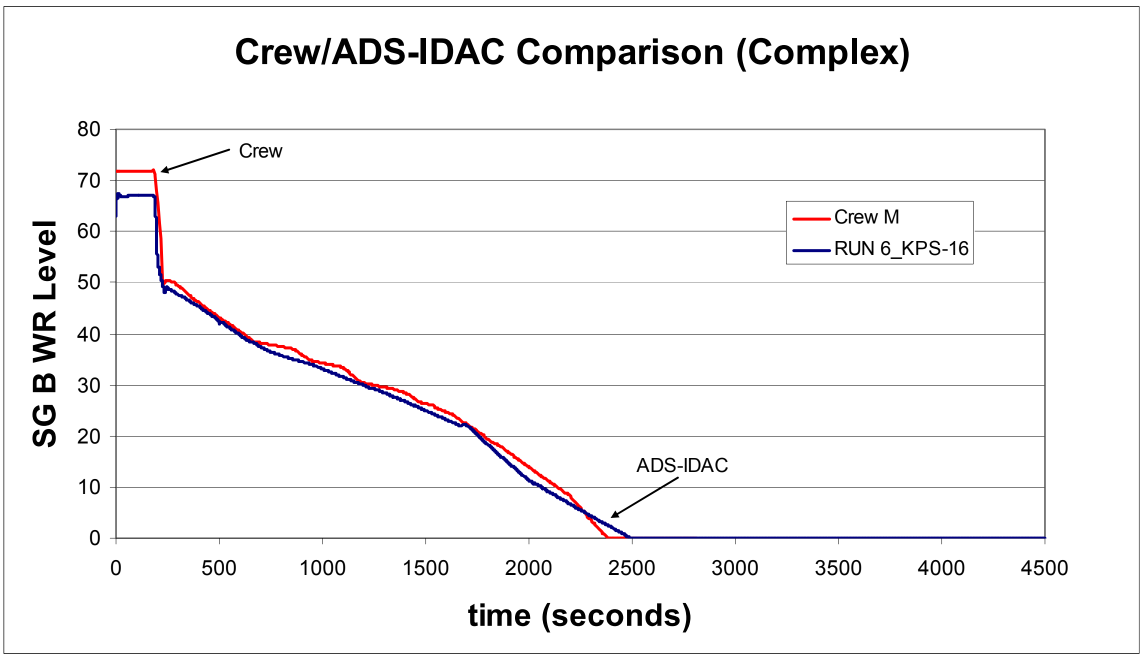


Figure 116 - Crew M (Complex LOFW) SG B Wide Range Water Level

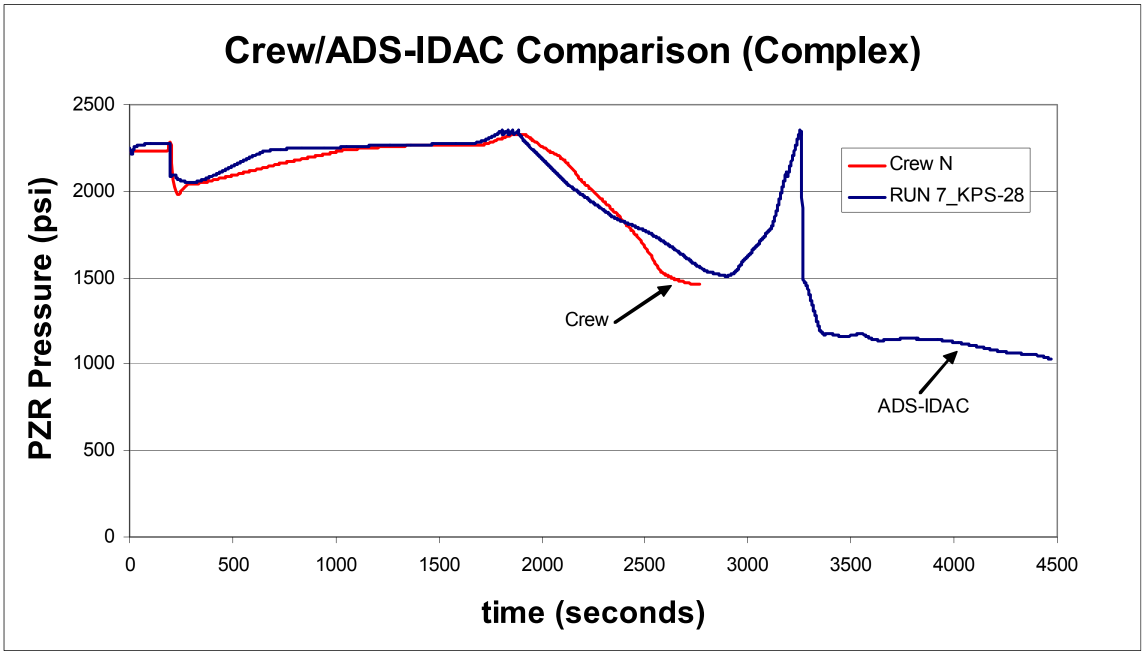


Figure 117 - Crew N (Complex LOFW) Pressurizer Pressure

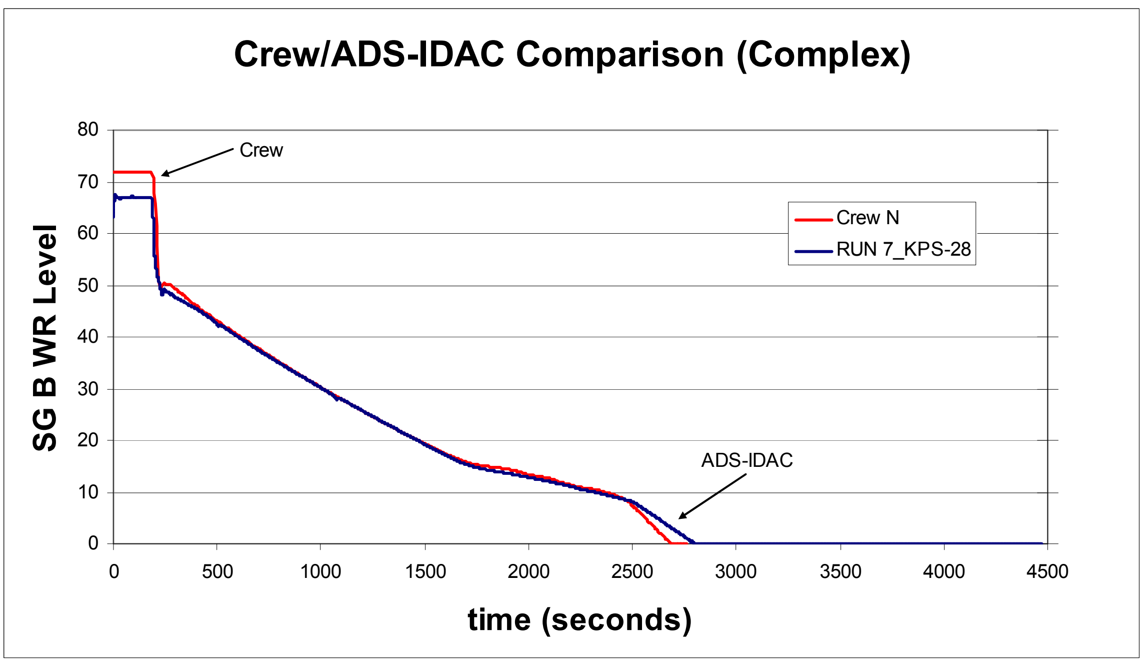


Figure 118 - Crew N (Complex LOFW) SG B Wide Range Water Level

(this page intentionally blank)

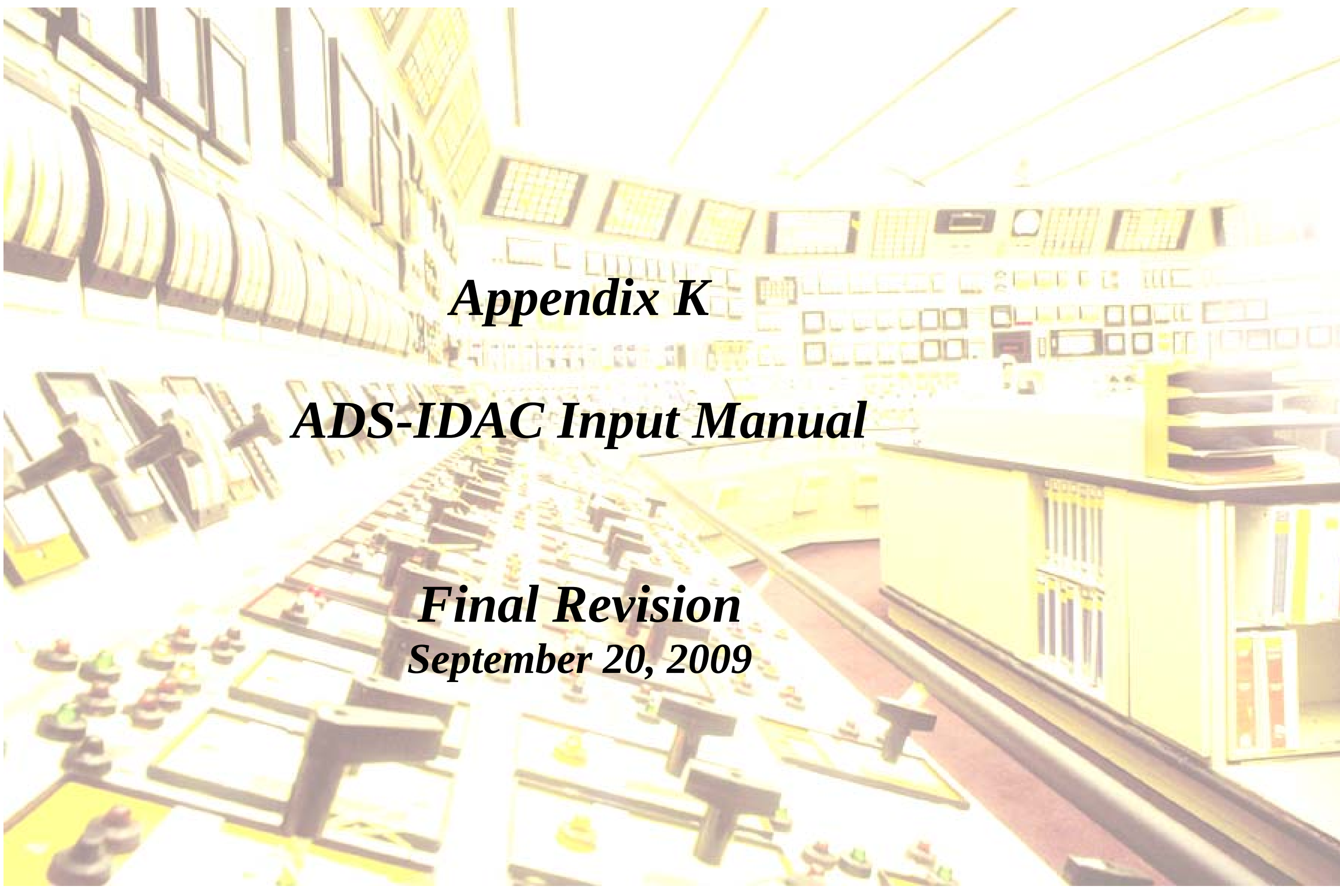
Appendix K – ADS-IDAC Input Manual

Appendix K contains the ADS-IDAC input manual. The input manual is intended to assist the analyst in developing the necessary information to execute the ADS-IDAC simulation code. The manual includes an overview of the program architecture (including detailed flowcharts for the implementation of each major phase of the IDAC model); general modeling guidance for developing an appropriate thermal-hydraulic nuclear plant model and identifying potential branching events; and detailed descriptions of each of the input files needed to run the code.

(this page intentionally blank)

Accident Dynamics Simulator
with the Information Decision Action Cognitive Model
in a Crew Context

ADS-IDAC



Appendix K

ADS-IDAC Input Manual

Final Revision

September 20, 2009

(this page intentionally blank)

Table of Contents

List of Figures

(this page intentionally blank)

Section 1: Introduction

- **ADS-IDAC Overview**

  Situational context and event dynamics are important considerations when assessing the potential for human errors. Dynamic probabilistic risk assessment methods can improve the prediction of human error events by providing rich contextual information and an explicit consideration of feedback arising from man-machine interactions. The Accident Dynamics Simulator paired with the Information, Decision, and Action in a Crew context cognitive model (ADS-IDAC) is a computational tool intended to support human reliability assessment within a dynamic simulation environment. A major goal of the ADS-IDAC project is the prediction of situational contexts that might lead to human error events, particularly knowledge driven errors of commission. The ADS-IDAC environment couples a thermal-hydraulic model with an operations crew cognitive model to permit the dynamic simulation of operator performance during nuclear power plant accidents. ADS-IDAC generates a discrete dynamic event tree using simple branching rules to model variations in crew responses. A significant advantage of the ADS-IDAC approach is the ability to directly assess the impact of operator actions on key plant parameters and more realistically determine the safety significance of human errors.

  The ADS-IDAC simulation code is still under development and model development, verification, and validation activities are ongoing. Although extensive debugging activities have been completed and error checking mechanisms have been added to the code, unexpected errors may still occur. Following the guidelines and instructions provided in this manual should minimize the likelihood of performance issues, but the user is urged to independently verify the results obtained from the code.

- **Input Manual Description**

  This manual is intended to provide the analyst with a reference manual for using the ADS-IDAC simulation code. Although a general overview of major modeling elements contained in the ADS-IDAC code is included in this manual, this information should not be considered to replace the detailed information contained in other sources (see Section 4: Reference List for additional information). The Input Manual is organized into four main sections:

  Section 1: Introduction- Provides a general overview of ADS-IDAC modeling elements, including flowcharts for key processes and general user guidance.

  Section 2: General Modeling Guidance- Provides guidance for the modeling potential error events and sources of crew-to-crew variability and development of a RELAP reactor plant model suitable for use with the ADS-IDAC code.

Section 3: ADS-IDAC File Structure - Provides a general overview of the data files that support execution of ADS-IDAC and a detailed description of the code output files.

Section 4: Input File Format - Provides detailed guidance on the content and format of the input files needed to run ADS-IDAC.

Section 5: Term Conversions – The current architecture of the ADS-IDAC source code utilizes a set of integer variables to identify key data attributes. For example, the source uses the integer variable VODM to designate the Decision Making Operator (e.g., a senior reactor operator). A term conversion table establishes the relationship between the variable name and the integer value (e.g., VODM = 3043). This allows key data descriptors to be represented by an integer variable which requires less computer memory while still permitting meaningful variable names. A complete listing of the integer term conversions is contained in this section.

Section 6: References: Provides a comprehensive reference list. Background information and detailed models relating to the ADS-IDAC code can be found in the listed references.

- **Input Manual Naming Conventions**

To improve the consistency and readability of this manual, several notation and naming conventions are used to identify variables, input files, and crew members:

- ➢ words in ***bold italics*** generally refer to input variables. Section 3 of this manual provides a description of the required format and recommended values for input variables.

- ➢ words surrounded by quotation marks (“”) generally refer to input file names. Section 3 provides a detailed description of all input files required to execute ADS-IDAC.

- ➢ words surrounded in straight brackets ([]) generally refer to optional or conditional parameters in the associated input file. These variables may not be needed depending on the content of the input file. The input file descriptions provided in Section 3 provides additional detail for handling these situations.

- ➢ ADS-IDAC currently models two crew members – a Decision Maker (representing a Shift Supervisor/Senior Reactor Operator) and an Action Taker (representing a Reactor Operator). Input files are associated with the Decision Maker and include the acronym “ODM” in file name, while files associated with the Action Taker include the acronym “OAT” (e.g., “KB_ODM_Event_Matrix.txt” and “KB_OAT_Event_Matrix.txt”).

- Input files associated with the operator knowledge base include the prefix "KB_" in the file name. Input files describing the content of procedure steps include the prefix "ZProcedure_".

- For consistency with customary probabilistic risk assessment practice, controls that can be manipulated by the operators include the prefix "X_". Alarm names are generally given the prefix "A_". Although this convention is not mandatory, it improves the readability of the input files.

Purpose:

These simplified process flowcharts are intended to provide a high level overview of the ADS-IDAC simulation program. Each flowchart consists of a high level block diagram followed by a detailed description of the processes associated with each block. In general, the flowcharts are intended to show the sequencing and context of the main simulation process steps. Process steps that rely on information contained in ADS-IDAC input files are highlighted.

Flowcharts:

ADS-IDAC Process Overview
Information Processing
Decision-Making Process (Action Taker)
Decision-Making Process (Decision Maker)
Goal Selection Process (Decision Maker)
Strategy Selection Process (Action Taker)
Strategy Selection Process (Decision Maker)
Follow Written Procedure Strategy Implementation
Knowledge-Based Reasoning Strategy Implementation

ADS-IDAC Process Overview

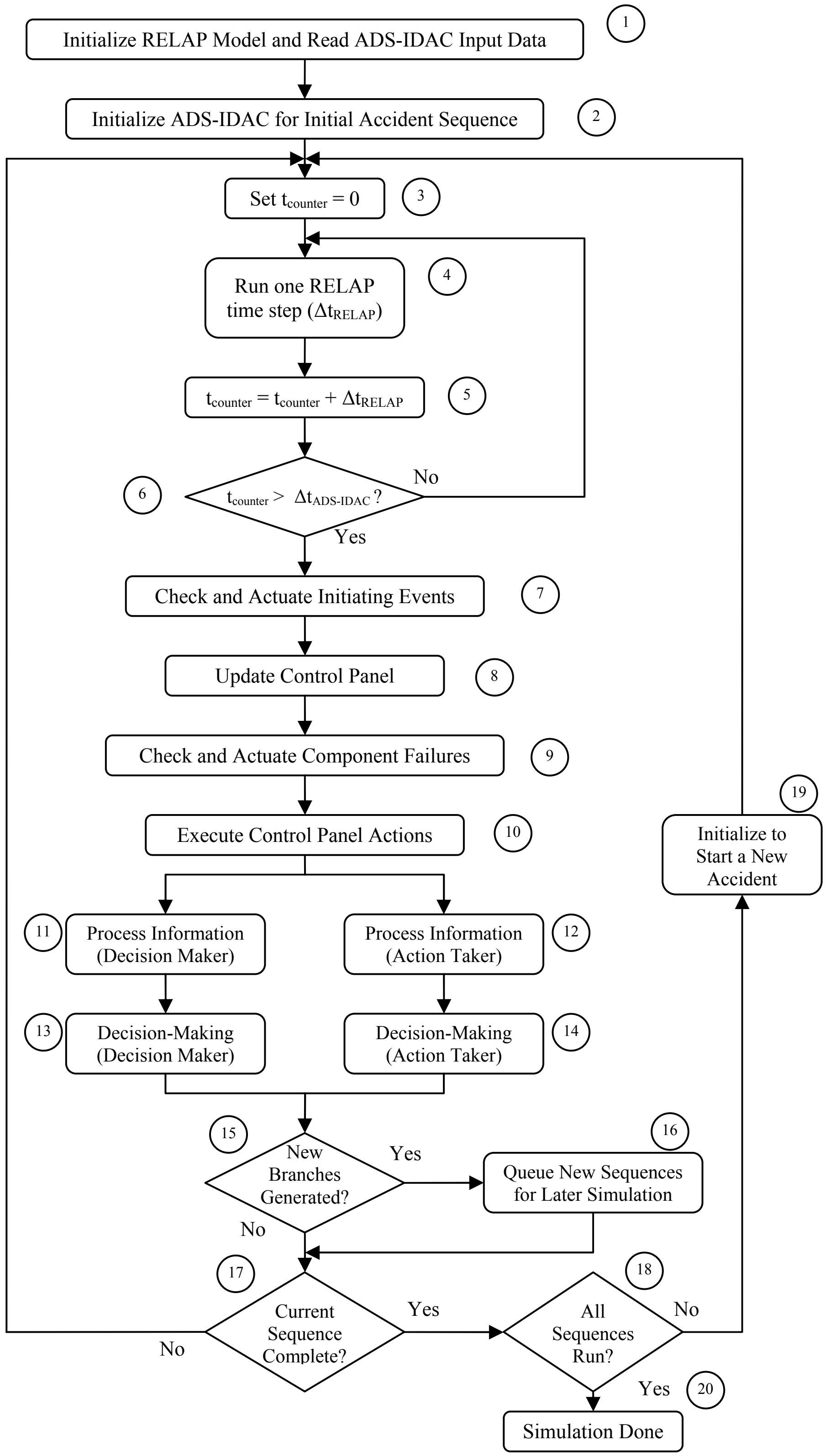


Figure 1, "ADS-IDAC Process Overview"

ADS-IDAC Process Overview Flowchart

Block 1 (Initialize RELAP Model and Read ADS-IDAC Input Data): This step initializes both the RELAP program and reads the ADS-IDAC input deck. The location of the RELAP input deck and the ADS-IDAC input files are specified in the ***ZiniADS.txt*** input file. This step includes several input file consistency checks including the presence of required input files, completeness of the system decomposition file (***KB_OAT(ODM)_System_Decomposition.txt***), and consistency in procedure step file names. However, due to the limited input file verification performed in the current code version, the user should be aware that successful execution of ADS-IDAC initialization does not necessarily indicate error free input. The user should refer to the log file ***err messages.txt*** to determine if errors have occurred during input file processing.

Block 2 (Initialize ADS-IDAC for Initial Accident Sequence): Initializes ADS-IDAC to begin the initial accident sequence (Sequence 0). Because a RELAP input deck error will result in immediate termination of the RELAP code, generation of the error message "all sequences simulated" during this step generally indicates an input deck error. Input deck errors can be identified by executing the RELAP code in standalone mode (i.e., outside the ADS_IDAC environment) and examining the resultant output file to identify syntax or formatting errors. The user should also refer to the log file ***err messages.txt*** to determine if other errors occurred during input file processing.

Block 3 (Initialize Counter): The ADS-IDAC scheduler provides supervisory control over the execution of the RELAP thermal hydraulic model. Consequently, execution of the ADS-IDAC code is controlled with two main time step parameters: (1) the internal RELAP5 time step, and (2) the ADS-IDAC time step. The RELAP5 time step (which must be equal to or less than the ADS-IDAC time step) establishes the incremental time step used in the reactor plant model thermal hydraulic calculation and is specified by Cards 200-299 in the RELAP input deck. The ADS-IDAC time step establishes the incremental time step used by the ADS-IDAC scheduler module. Each ADS-IDAC time step, the scheduler pauses the execution of the RELAP5 thermal hydraulic code in order to update operator model data, activate hardware failures and initiating events, and manipulate controls. Between these ADS-IDAC scheduler time step pauses, RELAP5 runs without interruption using the time step control specified in the RELAP input deck.

Block 4 (Run One RELAP Time Step): The RELAP5 thermal hydraulic model runs for one time step. The time step is established by the parameters provided in input deck cards 200-299.

Block 5 (Advance RELAP Counter): The accumulated time counter for RELAP5 execution is advanced by one time step.

Block 6 (Pause RELAP Run?): The accumulated time counter for RELAP5 is compared to the ADS-IDAC time step to determine if the RELAP5 code should be paused. If the accumulated time is less than the ADS-IDAC time step, another RELAP time step is executed, the counter is updated, and the comparison is reperformed. This process is repeated until the accumulated RELAP execution time exceeds the specified ADS-IDAC time step.

The default value for the ADS-IDAC time step is 0.5 seconds. However, the time step value may be changed by adjusting line 1505 of Fortran module ***dtstep.f***. This line reads:

if((timehy - timesno) .ge. ***ADS-IDAC_time_step****) then*

In addition, line 205 in the module ***CommandControlCenter.cpp*** must be set to the same time step value:

*this->reduceRemainingTime(****ADS-IDAC_time_step****)*

This ensures that the ADS-IDAC scheduler time step remains synchronized with the RELAP5.

Block 7 (Check and Actuate Initiating Events): The ADS-IDAC scheduler checks if any initiating events (specified in the ***Initiating_Event.txt*** input file) need to be activated.

Block 8 (Update Control Panel): All indicators listed in the ***ControlPanel.txt*** are updated based on the current RELAP thermal hydraulic data. It should be noted that simply updating control panel indicators does not automatically update the operator's perception of the plant state. The operator must first perceive and memorize information from the control panel in order to use the updated information.

Block 9 (Check and Activate Component Failures): The ADS-IDAC scheduler checks if any hardware reliability failures (specified in the ***SystemReliability.txt*** input file) need to be activated.

Block 10 (Execute Control Panel Actions): The ADS-IDAC scheduler checks if any control manipulation are awaiting execution. If there control manipulation have been activated, new control values are passed to the RELAP thermal hydraulic code through the interactive controls specified in the ***ControlPanel.txt*** and the ***RELAP5_channels.txt*** input files.

Block 11 (Process Information – Decision Maker): See the "Information Processing" flowchart for a description of this step.

Block 12 (Process Information – Action Taker): See the "Information Processing" flowchart for a description of this step.

Block 13 (Decision-Making Process – Decision Maker): See the "Decision-Making Process (Decision Maker)" flowchart for a description of this step.

Block 14 (Decision-Making Process – Action Taker): See the "Decision-Making Process (Action Taker)" flowchart for a description of this step.

Block 15 (Were Two or More New Branches Generated?): In ADS-IDAC, a branch is generated every time an event occurs. Events include changes in alarm states, hardware actuations or failures, control manipulations, operator decisions, and performance of procedure step activities. If two or more branches of the same branch type are generated during a time step, a new sequence path may be generated. In order to generate a new sequence, the following conditions must be met:

- The branches must be associated with one of the following branch types: (1) hardware reliability events, or (2) an operator related events associated with an action, procedure step expectation, goal, strategy, mental belief or information gathering.

- Two or more branches of the same branch type must be generated during the time step.

When a new sequence is generated, the events associated with each branch are partitioned among the new sequences. For example, if the Decision Maker generated two goal branches during a time step, the first goal branch is applied to one sequence and the second goal branch is applied to the other sequence. In the current version of ADS-IDAC, it is not possible to generate multiple sequences for more than one operator related branch type at a time. For this reason, the generation of operator related branch types is constrained to ensure that only one operator branch type can be generated during a time step (e.g., an operator is precluded from generating a goal and an action branch type during the same time step). When multiple branch event result in the generation of a new sequence, the ADS-IDAC scheduler will save the current operator crew and plant state information and queue the new sequence for later simulation.

Certain branch types such as changes in alarm states will not generate new sequences regardless of how many branches are activated during the time step. Furthermore, two or more branches of different branch types will not generate a new sequence (e.g., activation of a new goal branch for the Decision Maker and a new strategy branch for the Action Taker). In these cases, the events associated with each branch are applied to the current event sequence.

Block 16 (Queue New Accident Sequences for Later Simulation): When a new sequence is identified, the ADS-IDAC scheduler saves all information related to the current plant and operator states by either copying the information to computer memory (console version) or writing the information to the computer hard disk (multiple processor version). Information related to placement of the new sequence within the simulation Discrete Dynamic Event Tree is also saved (e.g., identity of parent sequence branch, number of new sequences generated, and events to be partitioned among the new sequences). New sequences are then placed in a holding queue to await later simulation.

Block 17 (Current Accident Sequence Complete?): The purpose of this step is to determine if any termination criteria have been met by the current sequence. In general, four criteria can be used to terminate a sequence: (1) the simulation time has reached the sequence the truncation time specified in the ***ZCtrlPar.txt*** input file, (2) the sequence executed the procedure non-response exit condition “VSTOP”, (3) the sequence reached the probability cutoff value specified in the ***ZCtrlPar.txt*** input file, or (4) an alarm condition with the name prefix “A_ENDSEQ” has been actuated. A sequence ca also be terminated by a RELAP thermal hydraulic error condition. However, a thermal hydraulic error precludes restart of a new sequence and therefore terminates the entire simulation.

Block 18 (Have All Accident Sequences Been Run?): Checks to determine if there are sequences waiting in the simulation queue. If all sequences have been simulated, the simulation ends.

Block 19 (Initialize ADS-IDAC to Restart with a New Accident Sequence): Based on information stored when a new sequence is generated (Block 16), the ADS-IDAC re-initializes the operator and plant models to restart a queued sequence. This includes re-initialization of the RELAP5 thermal hydraulic model back to the point when the new sequence was initially generated.

Block 20 (Simulation Done): Once all sequences have been simulated, the ADS-IDAC scheduler writes all pertinent simulation data to data output text files. The ADS-IDAC output includes the following files:

<u>ES File Folder</u>

- ES_summary.txt – Summarizes all sequences associated with the simulation. For each sequence, the sequence length, probability, and termination criteria are specified.
- Scenario info.txt – Provides a detailed description of all sequences, including all branching points and associated events.

<u>KPS File Folder</u>

KPSFolder***_folder_number*** **–** The key parameter state (KPS) output is arranged into individual file folders. ADS-IDAC limits the number of individual files within an output folder to less than 1000 files. Therefore, if more than 1000 output files are generated by a simulation, additional KPSFolders are created to store the data. The first 1000 output files are stored in KPSFolder_0, the seconds 1000 files are stored in KPSFolder_1, and so on. For each sequence, the following output files are created:

- ZDIA***_sequence_number***.txt – For each operator, provides the output from the diagnostic engine. Output consists of a time history of the diagnosis event confidence level for each operator.
- ZKPS***_sequence_number***.txt – Provides a time history of all indicators specified in the ***ControlPanel.txt*** input file.
- ZPIF***_sequence_number***.txt – For each operator, provides a time history of the dynamic PIF values for system criticality, time constrained loading, and information loading.

<u>PE File Folder</u>

PEFolder_***folder_number*** **–** The pivotal event (PE) output is arranged into individual file folders. A sequence consists of a series of connected PE nodes. Similar to the KPS file folder, ADS-IDAC limits the number of individual files within an output folder to less than 1000 files. Therefore, if more than 1000 output files are generated by a simulation, additional PEFolders are created to store the data. The first 1000 output files are stored in PEFolder_0, the seconds 1000 files are stored in PEFolder_1, and so on. The collection of PENode data files provides sufficient information to reconstruct the DDET for the simulation. For each PE node, the following output file is created:

- ZPENODE_***node_number***.txt – Specifies the type of branch event, the time of the event, and the sequence number associated with the PE node.

These output files can be imported into a third party program such as MS Excel for data analysis and visualization.

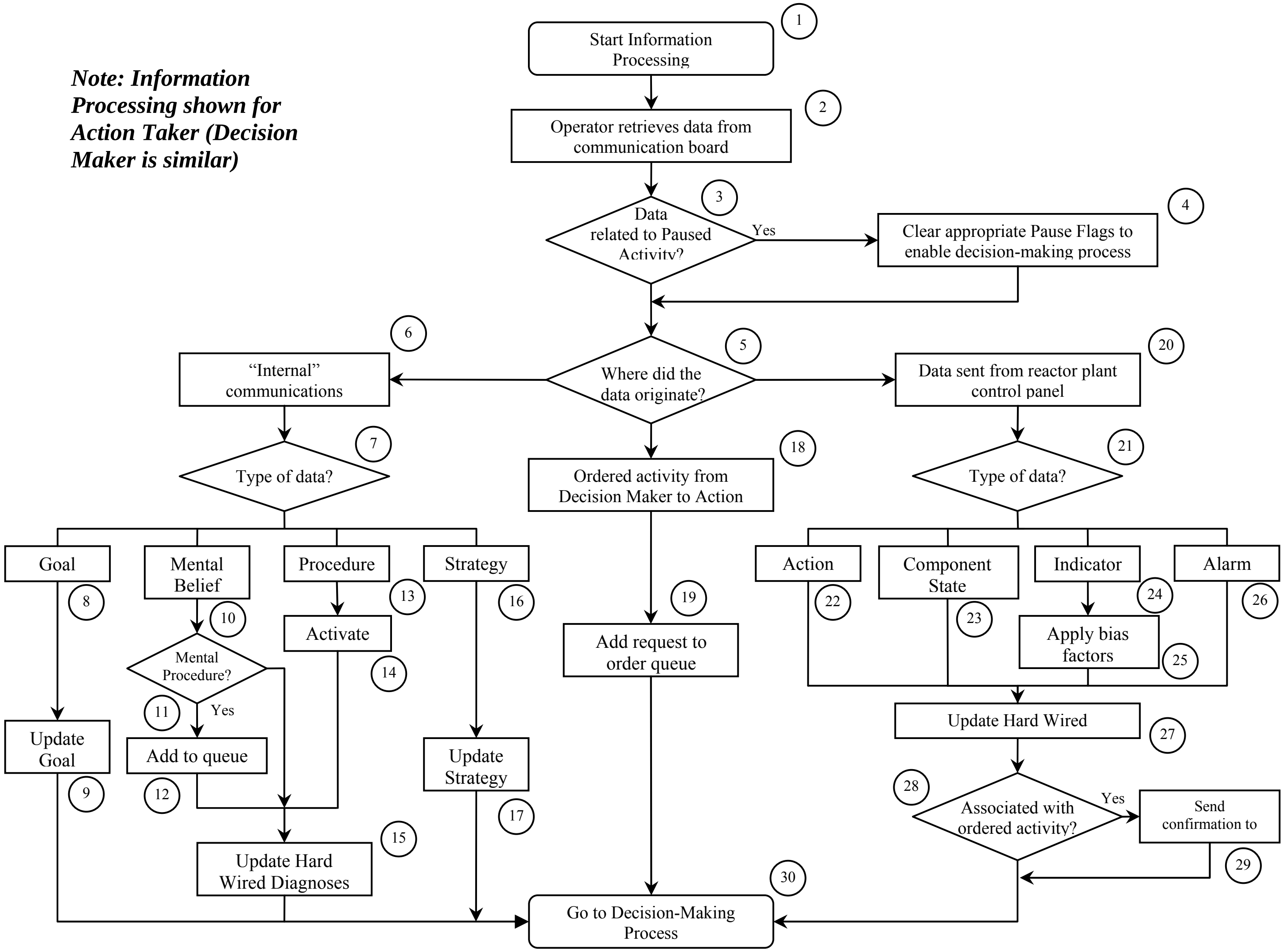


Figure 2, "Information Processing"

Information Processing Flow Chart – Description

Block 1 (Start Information Processing): The IDAC cognitive model consists of three main activities: information processing, decision-making, and action execution. Information processing is the cognitive activity in this process. In the ADS-IDAC model, the operator can only act upon perceived information. Therefore, raw information from the reactor plant model must first be gathered and memorized before the data can be used to support decision making. ADS-IDAC includes two information processing filters: (1) the control panel scanning queue, and the (2) bias factors. These filters are described in more detail in the ***KB_OAT(ODM)_Scanned_Parameters.txt*** and ***KB_OAT(ODM)_Bias_Factors.txt*** sections.

Block 2 (Retrieve Data from Communication Board): The ADS-IDAC employs central communication clearinghouse "blackboard" architecture for handling all information interchange. Communication items can be posted to the central blackboard by all members of the control room crew and the reactor plant. Each communication packet includes data identifying the sender, intended recipient, type of communication, and the content of the message. Each time step, the control room operators and the reactor plant scans the communication blackboard to identify their messages. For example, the reactor plant might post the actuation of an alarm to the blackboard with all operators identified as intended recipients. Similarly, when the reactor operator attempts to open a valve or start a pump, a message for the reactor plant is posted to the blackboard. Because the blackboard serves as a centralized dispatcher of all information exchange, it provides a convenient location for assessing the average information load for each operator.

Block 3 (Is Data Related to a Paused Activity): Initiation of certain operator activities cause the decision-making process to be temporarily suspended until the activity is completed. For example, if the Decision Maker requests the Action Taker to read a plant indicator, the Decision Maker will not proceed with further decision-making activities until the plant indicator status has been reported. This temporary suspension provides two important features: (1) it allows better modeling of the time required to execute certain operator actions, and (2) it allows better synchronization and coordination of crew activities. Decision-making activities are suspended by setting either of two operator "pause activity" Boolean flags to "true". One pause flag suspends decision-making activities associated with execution of memorized mental procedures while the other pause flag suspends all other operator activities (e.g., execution of written procedures, knowledge based actions, or goal and strategy selection). The appropriate pause flag is set to "true" whenever the operator begins an activity that is either associated with a time delay or requires crew coordination. When the operator receives a communication packet indicating that the activity associated with the pause is completed, the appropriate "pause activity" flag is set to false to enable the decision-making process. To support the control of pause activities, each communication packet includes data identifying if the communication is associated with a pause activity and, if applicable, the appropriate pause activity flag.

Block 4 (Clear Appropriate Pause Flags): If a received communication packet is associated with a pause activity, the operator's appropriate pause activity flag is set to "false" to enable the operator's decision-making process.

Block 5 (Identify the Data Sender/Originator): Communication packets can originate from three different sources: (1) internal communication from the operator himself/herself, (2) external crew communication from other operators, and (3) reactor plant status information from the control panel. The source of the communication establishes how further information processing is handled.

Block 6 (Internal Communications): Internal communication is used to transfer information related to cognitive decisions within the operator's own memory. For example, decisions about goals, strategies, and mental beliefs are only communicated internally within the operator.

Block 7 (Identify Type of Data): Internal communication is associated with four possible data types: goals, strategies, mental beliefs, or procedure steps. Each communication packet includes information that identifies the data type.

Block 8 (Goal Related Information): The information is related to a high level operator goal. Four possible goals are currently implemented in ADS-IDAC: (1) normal operation, (2) troubleshooting, (3) monitoring abnormal conditions, and (4) maintain global safety margin. When the current goal is changed, an internal communication packet is generated to update the goal in the operator's memory.

Block 9 (Update Goal): The operator updates perceived information memory to match the new goal. The high level operator goal impacts strategy selection during the decision-making process. Depending on the branching options selected for the simulation, a new goal may either be implemented or bypassed. If the goal is bypassed, the updating process is omitted and the new goal has no effect.

Block 10 (Mental Belief Related Information): The information is related to a mental belief. When an operator mental belief is activated (see ***KB_OAT(ODM)_HardWired_Diagnosis.txt)***, a communication packet is generated to add the mental belief to the operator's perceived information memory. Mental beliefs perform two main functions: (1) activation of associated mental procedures, and (2) prerequisite conditions for other mental beliefs. Similar to goals, branching options can be set for mental beliefs to either permit perception of the belief or to bypass the belief. If the mental belief is bypassed, the operator will not perceive the new mental belief.

Block 11 (Does Mental Belief Reference a Procedure): If the mental belief is associated with a memorized mental procedure, the associated mental procedure needs to be added to the operator's procedure queue.

Block 12 (Add Mental Procedure to Queue): Each operator maintains a memorized mental procedure queue to store mental procedures that are waiting to be executed. The procedure name, step number, priority level, and timing parameters are added to the mental queue during this step. During the decision-making process, the operator checks for the queue to determine if mental procedures are awaiting execution. If a queued mental procedure is found, a procedure related communication packet (see Blocks 13 and 14) is generated to activate the procedure. Once the procedure is activated, the operator will execute the associated actions.

Block 13 (Procedure Related Information): The information is related to a procedure. Proceduralized actions are initiated by the generation of a communication packet identifying the procedure name and step number of the desired action. Two types of procedures can be used in ADS-IDAC - a memorized mental procedure (generally associated with skill based memorized actions) or a conventional written procedure (e.g., emergency operating procedure). When a procedure related communication packet is generated, the associated procedure is activated. If an earlier active procedure was still in progress, the earlier procedure is placed in a "paused" status (i.e., the most recently activated procedure takes precedence).

Block 14 (Activate): The procedure is placed in an active status[1]. This enables the operator to begin implementation of the procedure actions during the decision-making process.

Block 15 (Update Hard Wired Diagnoses): Activation conditions for mental beliefs can include other mental beliefs and procedure status. Therefore, each mental belief in the operator knowledge base is updated based on the perceived mental belief and procedure information. See ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** for additional information.

Block 16 (Strategy Related Information): The information is related to a problem solving strategy. ADS-IDAC utilizes four strategies: wait and monitor; knowledge-base reasoning (Decision Maker only); follow written procedures (Decision Maker only), and follow instruction (Action Taker only). A strategy related communication packet is generated when the operator changes the current strategy. The strategy is selected during the decision-making process and is influenced by the crew's high level goals.

Block 17 (Update Strategy): The operator updates perceived information memory to match the new strategy. The current strategy establishes the problem solving methods used by the crew.

Block 18 (Ordered Activity from Decision Maker): In the ADS-IDAC crew model, Only the Action Taker is permitted to manipulate reactor plant controls. Furthermore,

---

[1] Procedures can be placed in any of four possible status categories:
NONE – the procedure is inactive and awaiting execution
ACTIVE – the procedure is in progress
PAUSE – the procedure was previously activated but was temporarily suspended
INTERRUPTED – the procedure has been permanently suspended
DONE – the procedure has been completed.

during execution of either a mental or written procedure, the Decision Maker requests the Action Taker to provide any required control panel information (unless use of memorized information is enabled and the Decision Maker has a valid memory of the required information). Therefore, if (1) the communication packet is associated with a control panel parameter, component state, alarm, or action; (2) the sender is the Decision Maker, and (3) the recipient is the Action Taker; the communication is associated with an ordered action. This step applies only to the Action Taker and is not applicable for the Decision Maker

Block 19 (Add Ordered Activity to Queue): If the Action Taker has received an ordered action form the Decision Maker, the requested action is placed in an ordered action queue. . During the decision-making process, the Action Taker checks for queued ordered actions that have not been initiated. If an ordered action is found, the Action Taker will interact with the reactor plant to either perform the requested action or obtain the needed information.

Block 20 (Information From Reactor Plant): When an operator executes interactions with the control panel, a communication packet is sent to the reactor plant. Control panel interactions include control manipulations (e.g., opening a valve, stopping a pump), checking plant parameters, and verification of component or alarm states. Upon receipt of the communication packet, reactor plant executes the interaction and sends a communication packet back to the operator to confirm completion of the requested action or provide requested plant data.

Block 21 (Identify Type of Information): The reactor plant control panel can send four types of information: action confirmation, component states, parameter values (indicators), and alarms. The reactor plant can initiate communications related to alarms (e.g., actuation of reactor trip alarm), but other information types are generated as a result of an operator control panel interaction. Communication packets initiated by the control panel provide two key functions: (1) the time taken by real operators to perform control panels action can be modeled by adjusting the timing of control panel communication, and (2) the communication packet provides a controlled conduit for sending plant data to a specific operator.

Block 22 (Action Related Information): After the control panel executes a control manipulation, a communication packet is sent back to the operator who initiated the request to confirm that the action has been completed.

Block 23 (Component State Information): The communication packet provides the operator with requested component state information.

Block 24 (Indicator Information): The communication packet provides the operator with requested parameter value information. The parameter value may be subsequently adjusted by a bias factor (see Block 25).

Block 25 (Apply Bias Factor): If bias factors are identified (see ***KB_OAT(ODM)_Bias_Factors.txt***), the parameter value is adjusted appropriately. The operator will perceive the biased indicator value rather than the value initially provided by the reactor plant control panel. The use of bias factors allows the code user to model certain types of indicator failures.

Block 26 (Alarm Information): The communication packet provides the operator with either requested parameter value information or new alarms. New alarms are self initiated by the control panel and include both the actuation and clearance of alarm conditions.

Block 27 (Update Hard Wired Diagnoses): Activation conditions for mental beliefs can include other control manipulations, alarms, component states, and parameter values. Therefore, each mental belief in the operator knowledge base is updated based on the perceived information. See ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** for additional information.

Block 28 (Was Information From Reactor Plant Associated With an Ordered Activity?): The ordered action queue is checked to determine if the perceived information is associated with an activity ordered by the Decision Maker. This step is only applicable to the Action Taker.

Block 29 (Send Ordered Activity Confirmation to Decision Maker): A communication packet is sent from the Taker to the Decision Maker to confirm completion of an ordered action. When the Decision Maker initiates an ordered action, further decision-making processing is paused until the Decision Maker receives confirmation that the action has been completed.

Block 30 (Proceed to Decision-Making Process): After information processing is complete, the operator initiates the decision making process.

Decision-Making Process (Action Taker)

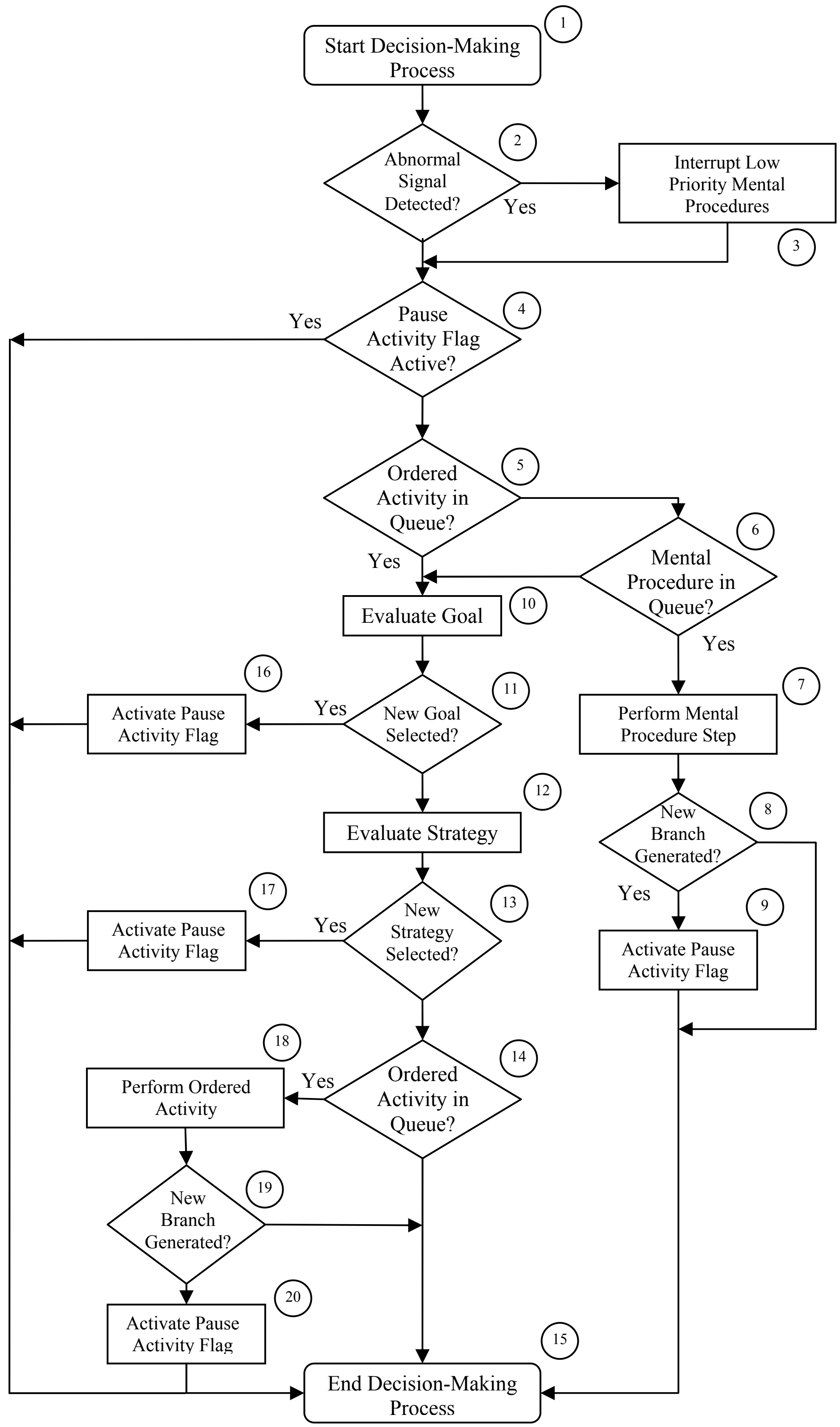


Figure 3, "Decision-Making Process (Action Taker)"

Decision-Making Process (Action Taker)

Block 1 (Start Decision-Making Process): The decision-making process immediately follows information processing. During the decision-making process, the operator evaluates the current goal and strategy, identifies actions and other activities in accordance with the selected problem-solving strategy, and implements memorized mental procedures.

Block 2 (Abnormal Signal Detected?): An abnormal signal indicates a condition that is incompatible with continued power operation. Either of two conditions are used to identify an abnormal signal: (1) activation of the "Reactor_Tripped" mental belief (See ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** for additional information), or (2) the fuzzy logic diagnostic engine accident relationship value exceeding a preset threshold (see ***Action_Taker.txt*** and ***KB_OAT_Event_Matrix.txt*** for more information on the diagnostic engine). Detection of an abnormal condition also shifts the control room crew into an accident response mode that can lead (depending on the operator preferences and tendencies selected in the ***Action_Taker.txt*** and ***Decision_Maker.txt*** input files) to implementation of the emergency operating procedures.

Block 3 (Interrupt Low Priority Mental Procedures): Because certain memorized mental procedures are incompatible with a post-trip reactor plant condition, certain low priority mental procedures are permanently suspended once an abnormal condition is detected. For example, a mental procedure intended to reduce turbine load is no longer appropriate once the reactor and turbine have tripped. The interruption of low priority procedures can occur only once during a simulation – once an abnormal signal has been detected, newly identified low priority mental procedures can be added to the operator queue and implemented. The low priority procedure threshold is established in the ***Action_Taker.txt*** input file. The priority level for a mental procedure is included in the mental belief input data in the ***KB_OAT_HardWired_Diagnosis.txt*** input file (lower priority values indicate higher priority procedures).

Block 4 (Pause Activity Flag Active?): The operator pause activity flags synchronize crew actions and control the pacing of operator activities. Each operator has two separate Boolean pause activity flags – one associated with mental procedures and another associated with all other operator activities. Because each operator is capable of performing only one activity at a time, the pause flags prevent overloading the operator with multiple activities. An active pause activity flag (i.e., flag to "true"), indicates that the operator is already performing an activity. Therefore, further decision-making activities are bypassed.

Block 5 (Ordered Activity in Queue?): If the Action Taker has been assigned an ordered activity by the Decision Maker, execution of memorized mental procedures is bypassed. This allows actions ordered by the Decision Maker to take priority over memorized actions that are self-identified by the Action Taker.

Block 6 (Mental Procedure in Queue?): If there are no waiting ordered activities, the mental procedure queue is checked to determine if any mental procedures are waiting to be executed. Due to limitations in the ADS-IDAC procedure modeling process, it is not possible for two or more operators to simultaneously perform the same procedure step. Therefore, a check is also made during this step to ensure that the Decision Maker is not performing the same mental procedure. If a waiting mental procedure exists (and is not being performed by a different operator), execution of the mental procedure takes priority over other decision-making activities such as goal and strategy selection.

Block 7 (Perform Mental Procedure Step): The operator executes the mental procedure actions.

Block 8 (New Branch Generated?): Some procedure actions require the operator to interact with the reactor plant or coordinate with other crew members. Typical examples of these tasks include checking the status of plant equipment, manipulating a control, or verifying a parameter value. Because these tasks require some amount of time to complete, further operator decision-making process is suspended until the task is complete. Within the ADS-IDAC, the generation of these tasks is accomplished by creating a new branch event. A branch represents a discrete event or information chunk created during a simulation sequence. Branch generation in this context should be distinguished from the creation of branches in a discrete dynamic event tree (DDET). Most branches do not cause multiple accident sequences to be generated. However, when two or more branches of the same type are generated in a single time step, ADS splits the current sequence path into two or more separate paths. The splitting of a sequence into two or more sequence paths corresponds to the generation of a DDET branching event.

Block 9 (Activate Pause Activity Flag): If a new branch is generated, the appropriate operator pause activity flag is activated (i.e., set to “true”). Further operator decision-making is suspended until the pause activity flag is reset during later information processing. This allows ADS-IDAC to model the time required to perform tasks and activities and prevents overloading the operator with multiple simultaneous tasks.

Block 10 (Evaluate Goal): Evaluate appropriateness of current high level operator goal. During this step, the Action Taker’s high level goal is compared to the Decision Maker’s high level goal. If the two goals do not match, the Action Taker’s goal is updated to match the Decision Maker’s goal.

Block 11 (New Goal Selected?): If the result of the goal evaluation process indicates that the operator’s current goal must be changed, further decision-making is suspended and the high level goal is updated.

Block 12 (Evaluate Strategy): Evaluate appropriateness of current problem solving strategy. See Strategy Selection Process (Action Taker) flowchart.

Block 13 (New Strategy Selected?): If the result of the strategy evaluation process indicates that the operator's current strategy must be changed, further decision-making is suspended and the problem solving strategy is updated.

Block 14 (Ordered Activity in Queue?): The ordered action queue is checked to identify activities previously ordered by the Decision Maker that are awaiting execution. If an ordered action is found, the Action Taker performs the action. The ordered action queue is a first in-first out queue (i.e. the oldest ordered action is performed first).

Block 15 (End Decision-Making Process): End of decision-making process.

Block 16 (Activate Pause Activity Flag): If the high level goal will be updated, a new branch is generated for the goal and the operator pause activity flag is activated to prevent further decision-making activities. When the new goal is perceived during subsequent information processing, the pause activity flag is cleared and decision-making activities resume.

Block 17 (Activate Pause Activity Flag): If the problem solving strategy will be updated, a new branch is generated for the strategy and the operator pause activity flag is activated to prevent further decision-making activities. When the new strategy is perceived during subsequent information processing, the pause activity flag is cleared and decision-making activities resume.

Block 18 (Perform Ordered Activity): If a previously ordered activity is found, the Action Taker performs the requested action.

Block 19 (New Branch Generated?): The execution of certain ordered actions will require the generation of a new branch. For example, actions associated with a time delay or that require interaction with the control panel or another operator will require that a new branch is generated.

Block 20 (Activate Pause Activity Flag): If the ordered activity results in the generation of a new branch, the operator pause activity flag is activated to prevent further decision-making activities. When the operator receives confirmation from the reactor plant control panel that the ordered action has been completed during a subsequent information processing, the pause activity flag is cleared and decision-making activities resume.

Decision-Making Process (Decision Maker)

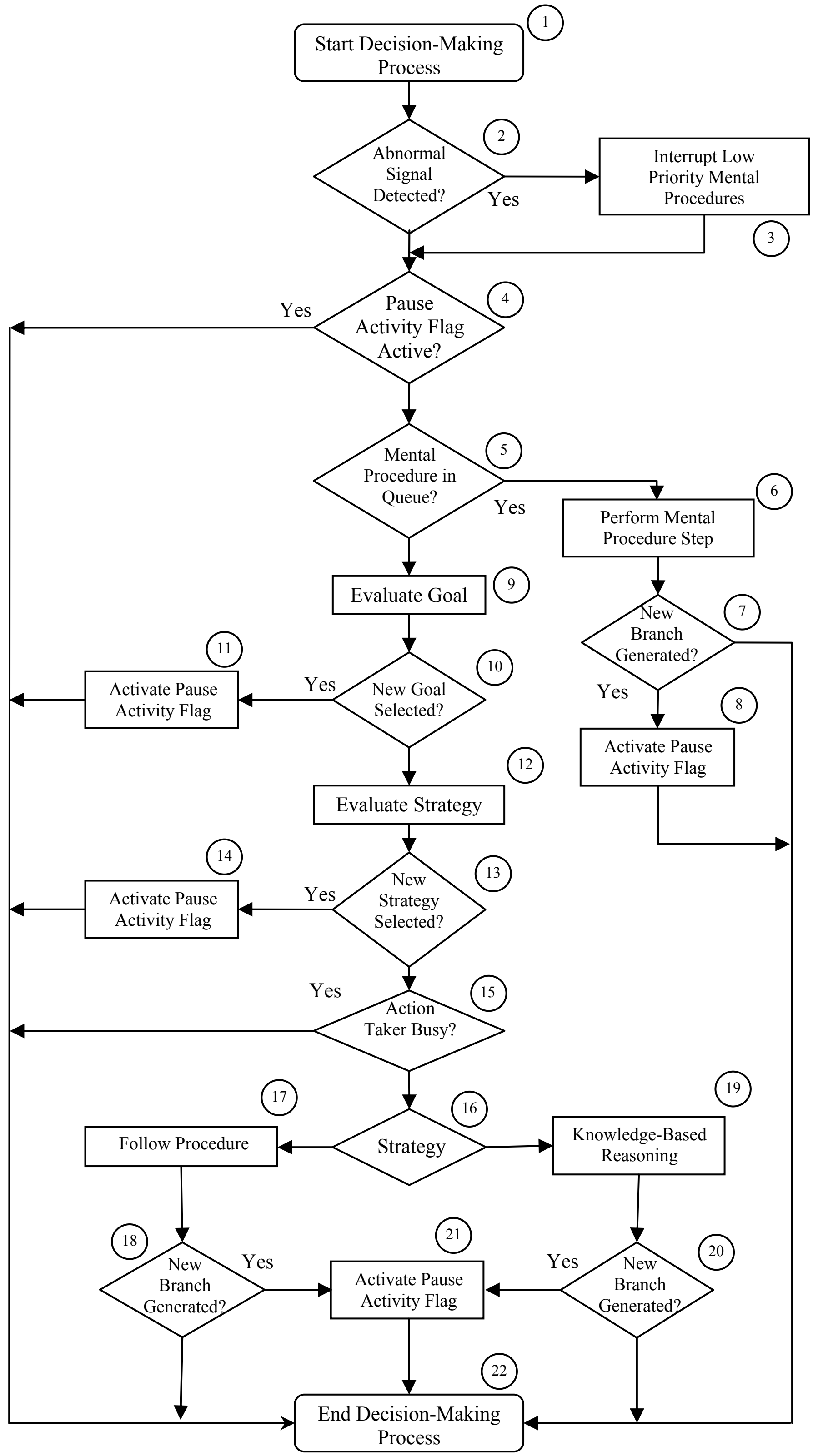


Figure 4, "Decision-Making Process (Decision Maker)"

Decision-Making Process (Decision Maker)

Block 1 (Start Decision-Making Process): The decision-making process immediately follows information processing. During the decision-making process, the operator evaluates the current goal and strategy, identifies actions and other activities in accordance with the selected problem-solving strategy, and implements memorized mental procedures.

Block 2 (Abnormal Signal Detected?): An abnormal signal indicates a condition that is incompatible with continued power operation. Either of two conditions are used to identify an abnormal signal: (1) activation of the "Reactor_Tripped" mental belief (See ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** for additional information), or (2) the fuzzy logic diagnostic engine accident relationship value exceeding a preset threshold (see ***Action_Taker.txt*** and ***KB_OAT_Event_Matrix.txt*** for more information on the diagnostic engine). Detection of an abnormal condition also shifts the control room crew into an accident response mode that can lead (depending on the operator preferences and tendencies selected in the ***Action_Taker.txt*** and ***Decision_Maker.txt*** input files) to implementation of the emergency operating procedures.

Block 3 (Interrupt Low Priority Mental Procedures): Because certain memorized mental procedures are incompatible with a post-trip reactor plant condition, certain low priority mental procedures are permanently suspended once an abnormal condition is detected. For example, a mental procedure intended to reduce turbine load is no longer appropriate once the reactor and turbine have tripped. The interruption of low priority procedures can occur only once during a simulation – once an abnormal signal has been detected, newly identified low priority mental procedures can be added to the operator queue and implemented. The low priority procedure threshold is established in the ***Decision_Maker.txt*** input file. The priority level for a mental procedure is included in the mental belief input data in the ***KB_ODM_HardWired_Diagnosis.txt*** input file (lower priority values indicate higher priority procedures).

Block 4 (Pause Activity Flag Activated?): The operator pause activity flags synchronize crew actions and control the pacing of operator activities. Each operator has two separate Boolean pause activity flags – one associated with mental procedures and another associated with all other operator activities. Because each operator is capable of performing only one activity at a time, the pause flags prevent overloading the operator with multiple activities. An active pause activity flag (i.e., flag to "true"), indicates that the operator is already performing an activity. Therefore, further decision-making activities are bypassed.

Block 5 (Mental Procedure in Queue?): If there are no waiting ordered activities, the mental procedure queue is checked to determine if any mental procedures are waiting to be executed. Due to limitations in the ADS-IDAC procedure modeling process, it is not possible for two or more operators to simultaneously perform the same procedure step. Therefore, a check is also made during this step to ensure that the Action Taker is not performing the same mental procedure. If a waiting mental procedure exists (and is not being performed by a different operator), execution of the procedure takes priority over other decision-making activities such as goal and strategy selection.

Block 6 (Perform Mental Procedure Step): The operator executes the mental procedure actions.

Block 7 (New Branch Generated?): Some procedure actions require the operator to interact with the reactor plant or coordinate with other crew members. Typical examples of these tasks include checking the status of plant equipment, manipulating a control, or verifying a parameter value. Because these tasks require some amount of time to complete, further operator decision-making process is suspended until the task is complete. Within the ADS-IDAC, the generation of these tasks is accomplished by creating a new branch event. A branch represents a discrete event or information chunk created during a simulation sequence. Branch generation in this context should be distinguished from the creation of branches in a discrete dynamic event tree (DDET). Most branches do not cause multiple accident sequences to be generated. However, when two or more branches of the same type are generated in a single time step, ADS splits the current sequence path into two or more separate paths. The splitting of a sequence into two or more sequence paths corresponds to the generation of a DDET branching event.

Block 8 (Activate Pause Activity Flag): If a new branch is generated, the appropriate operator pause activity flag is activated (i.e., set to "true"). Further operator decision-making is suspended until the pause activity flag is reset during later information processing. This allows ADS-IDAC to model the time required to perform tasks and activities and prevents overloading the operator with multiple simultaneous tasks.

Block 9 (Evaluate Goal): Evaluate appropriateness of current high level operator goal. See Goal Selection Process Flowchart (Decision Maker).

Block 10 (New Goal Selected?): If the result of the goal evaluation process indicates that the operator's current goal must be changed, further decision-making is suspended and the high level goal is updated.

Block 11 (Activate Pause Activity Flag): If the high level goal will be updated, a new branch is generated for the goal and the operator pause activity flag is activated to prevent further decision-making activities. When the new goal is perceived during subsequent information processing, the pause activity flag is cleared and decision-making activities resume.

Block 12 (Evaluate Strategy): Evaluate appropriateness of current problem solving strategy. See Strategy Selection Process (Decision Maker) flowchart.

Block 13 (New Strategy Selected?): If the result of the strategy evaluation process indicates that the operator's current strategy must be changed, further decision-making is suspended and the problem solving strategy is updated.

Block 14 (Activate Pause Activity Flag): If the problem solving strategy will be updated, a new branch is generated for the strategy and the operator pause activity flag is activated to prevent further decision-making activities. When the new strategy is perceived during subsequent information processing, the pause activity flag is cleared and decision-making activities resume.

Block 15 (Action Taker Busy?): In order execute the results of further decision-making activities, it is likely that that the Decision Maker will need to issue orders to the Action Taker. To prevent overloading the Action Taker, a check is made to determine if the Action Taker is busy. If the Action Taker has a queued mental procedure waiting to be executed or has any pause activity flag activity flag activated (indicating that another activity is already in progress), the Action Taker is presumed to be busy and further Decision Maker decision-making activities is bypassed.

Block 16 (Current Strategy?): Depending on the Decision Maker's current problem solving strategy, the decision-making process follows three possible paths. If the current strategy is "Follow Written Procedure", the Decision Maker proceeds to block 17, "Follow Procedure". If the current strategy is "Knowledge-Based Reasoning", the operator proceeds to block 19. If the strategy is "Wait & Monitor" the operator bypasses further decision-making activities (this path is not shown on the flowchart).

Block 17 (Follow Written Procedure Strategy): The Decision Maker implements the follow procedure strategy (see the Procedure Following flowchart).

Block 18 (New Branch Generated?): Certain procedure following actions require either (1) interaction with another operator or the reactor plant control panel, or (2) are associated with a time delay. In these cases, a new branch is generated to temporarily suspend further decision-making until the action can be completed.

Block 19 (Knowledge-Based Reasoning Strategy): The Decision Maker implements the Knowledge-Based Reasoning strategy (see the Knowledge-Based Action Execution flowchart).

Block 20 (New Branch Generated?): Certain knowledge-based reasoning actions require either (1) interaction with another operator or the reactor plant control panel, or (2) are associated with a time delay. In these cases, a new branch is generated to temporarily suspend further decision-making until the action can be completed.

Block 21 (Activate Pause Activity Flag): When a new branch is generated, the pause activity flag is activated to temporarily suspend further decision-making activities until the associated activity is completed.

Block 22 (End Decision-Making Process): End of decision-making process.

Goal Selection Process (Decision Maker)

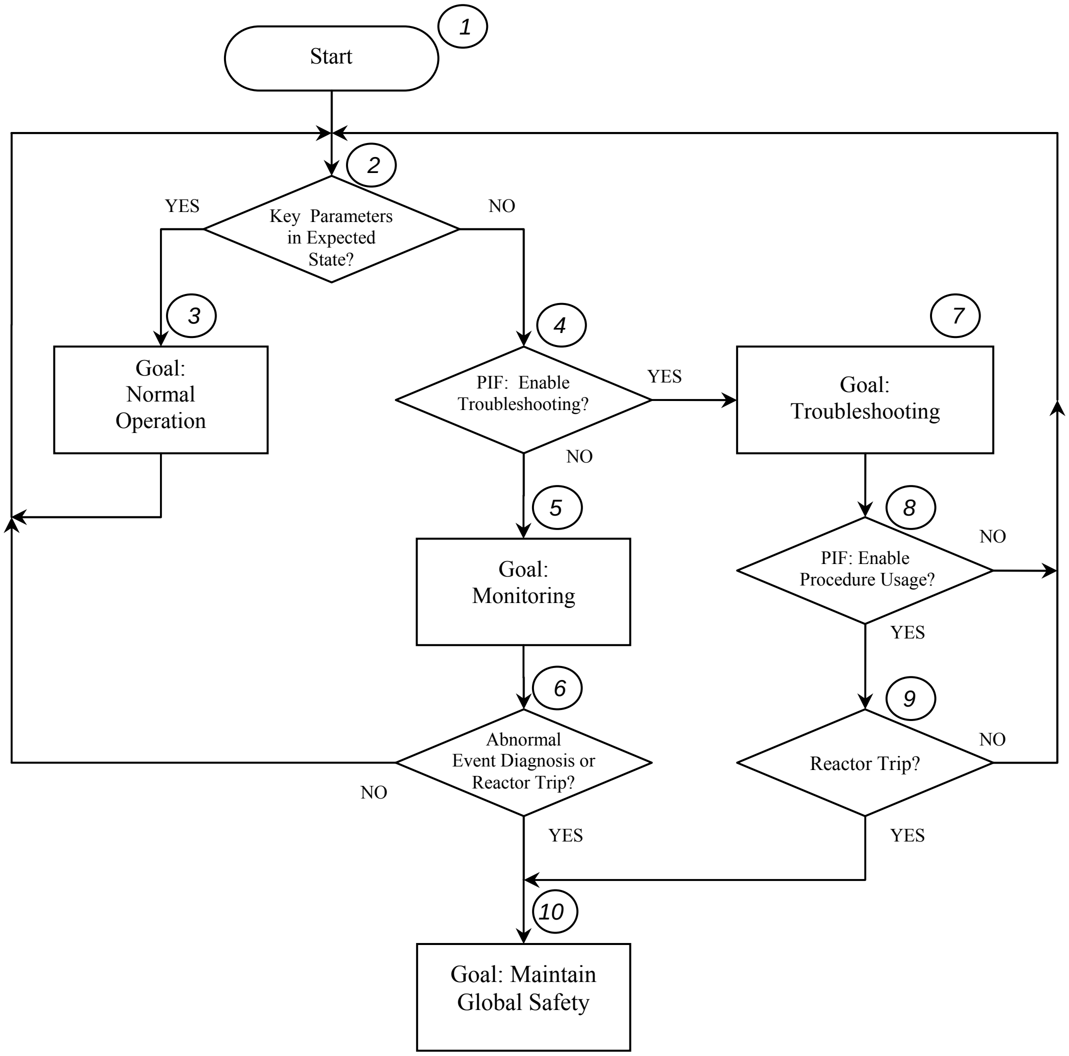


Figure 5, "Goal Selection Process (Decision Maker)"

Decision Maker[2] Goal Selection Flowchart Description

During each simulation time-step, the operators re-evaluate their high level goal. The goal evaluation and selection process includes the following main activities:

Block 1 (Start): Small errors in the nuclear plant model RELAP input deck initial conditions may cause minor plant transients that are not indicative of an actual abnormal condition. These transient conditions may actuate spurious alarms or other abnormal indications, but usually decay within the first several minutes of a simulation run. To prevent these artificial perturbations from influencing operator decision-making, Operator goal selection is blocked for an predetermined "dead time" after the start of the simulation to permit the RELAP thermal-hydraulic model to reach a steady-state condition.

Block 2 (Normal Operation Mental Belief): The operator knowledge base includes a "Normal Operation" mental belief that describes of the plant parameters associated with a normal full power operating condition. Although this description can be customized to reflect variabilities in operator knowledge, a typical normal operating condition description for a pressurized water reactor might include the following elements:

- Reactor Coolant System Temperature
- Reactor Coolant System Pressure
- Pressurizer Level
- Steam Generator Water Levels

The analyst should specify the expected values (and trends if applicable) for these parameters during normal plant operating conditions. Additionally, the analyst can adjust a threshold parameter that establishes how many of the specified conditions must be met in order for the operator to conclude that the plant is in a normal operating state. If a sufficient number of specified conditions are satisfied, the operator will initiate (or maintain) a mental belief that the plant is in normal operation.

Block 3 (Goal: Normal Operation): If the operator has a current mental belief that the plant is in a normal operating state, the goal state will be set to "Normal Operation". A goal of normal operation drives the following operator behaviors:

- Passive information gathering
- Control Panel Scanning
- Rule-Based Memorized Actions

Knowledge based problem solving and procedure following strategies are blocked when the goal is normal operation.

Block 4 (Enable Troubleshooting): The operator profile includes a static performance influencing factor that reflects the operators tendency to delay initiation of the emergency operating procedures in order to troubleshoot the perceived abnormal condition. The enable troubleshooting PIF can be set in the range on 0.0 – 1.0 to reflect operator

[2] A goal selection process flowchart is provided for the Decision Maker only. The Action Taker Goal Selection process ensures that the Action Taker mirrors the Decision Maker's goal. Therefore, the Action Taker only compares their goal to the Decision Maker goal and initiates a goal update when needed.

behavior. A value of 0.0 blocks troubleshooting activities while a value of 1.0 enables troubleshooting behavior and blocks activation of the "Monitoring" goal state. For intermediate values between 0.0 and 1.0, a branching point is generated to allow simulation of either option. The branching probability is equal to the enable troubleshooting PIF factor.

Block 5 (Monitoring Goal): The Monitoring goal is entered when the operator perceives that the plant is no longer in a normal operating state but a specific abnormal condition has not been detected. The Monitoring goal serves two main functions: (1) enable later activation of the Maintain Global Safety Margin goal, and (2) permit the operator perform active and passive information gathering activities in order to diagnosis the abnormal condition. Should plant parameters return to normal values, the operator may return to the Normal Operation goal.

Block 6 (Abnormal Condition Detected): Based on perceived information, the operator's fuzzy logic diagnostic engine generates a spectrum of possible abnormal and emergency conditions that might have occurred. When the membership values calculated by the fuzzy logic diagnosis process exceed a predetermined threshold (set in the operator profile), the operator concludes that an abnormal condition has occurred. In addition to the diagnostic engine output, detection of a reactor trip condition will also activate an abnormal condition operator belief.

Block 7 (Goal: Troubleshooting): Activation of the troubleshooting goal enables the reasoning based "Knowledge-base Problem Solving" strategy. Should the reactor plant return to a normal operating state, the operator may return to the Normal Operation goal. Once the operator activates the Troubleshooting goal, the fuzzy logic diagnostic engine will no longer influence operator behavior. The only conditions that will cause a transition from the Troubleshooting goal are: (1) the restoration of normal plant operation, or (2) a reactor shutdown.

Block 8 (Enable Procedure Usage): The operator profile includes a static PIF factor that reflects the operator's tendency to transition from the knowledge-based reasoning troubleshooting strategy to the written procedure following strategy. The enable procedure usage PIF factor ranges from 0.0 to 1.0. If the PIF factor is set to 0.0, the transition to procedure following is blocked and the operator will continue implementation of the troubleshooting strategy. If the factor is set to 1.0, once the operator perceives that the reactor has tripped, the troubleshooting strategy will be abandoned in favor of following written procedures. For intermediate PIF values between 0.0 and 1.0, a branching point is generated to allow simulation of either option. The branching probability is equal to the enable troubleshooting PIF factor.

Block 9 (Goal: Maintain Global Safety Margin): Activation of the Maintain Global Safety Margin goal blocks further implementation of knowledge-based action (if the Troubleshooting goal had been active) and allows execution of the "Follow Written Procedure" problem solving strategy. Once the Maintain Global Safety Margin goal is activated, transitions to other goal states are prohibited (i.e., the operator may not return to either the Normal Operation, Monitoring, or Troubleshooting goals).

Strategy Selection Process (Action Taker)

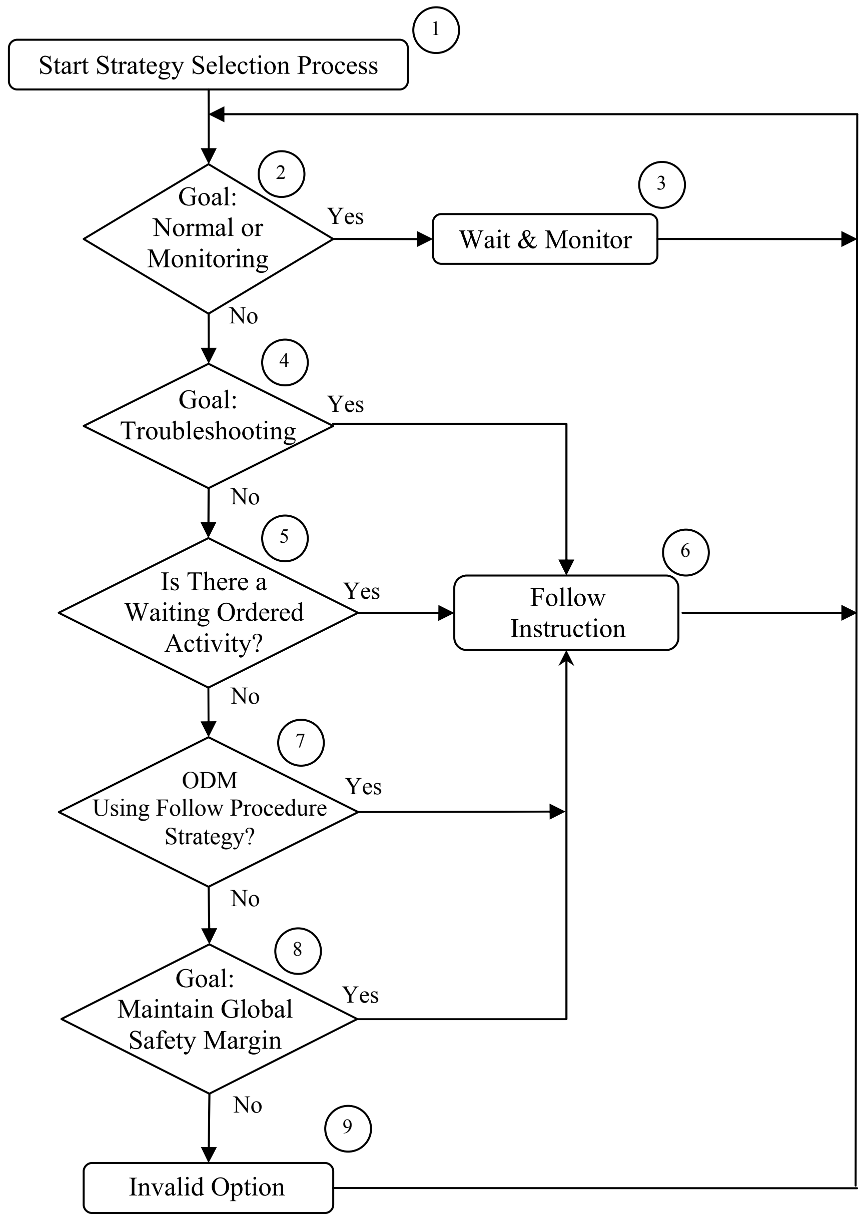


Figure 6, "Strategy Selection Process (Action Taker)"

Strategy Selection Process (Action Taker)

The Action Taker uses only two strategies: (1) Wait and Monitor, and (2) Follow Instruction. The Wait & Monitor strategy is an information gathering problem solving approach intended to improve the operator's situational assessment. The Follow Instruction strategy is activated when the Action Taker has received (or is expected to receive) an action order from the Decision Maker.

Block 1 (Start Strategy Selection Process): The strategy selection process executed each time the operator performs the decision-making step. The objective of strategy selection is to identify an appropriate problem solving strategy based on the operator's high level goal and preferences.

Block 2 (Is Goal Normal Operation or Monitoring?): Is the goal to maintain normal operation or to monitor an off-normal situation?

Block 3 (Wait and Monitor Strategy): Activate the "Wait and Monitor" problem solving strategy. This strategy involves passive information gathering and, if applicable, the execution of mental procedures.

Block 4 (Is Goal Troubleshooting?): Is the current high level goal to troubleshoot an abnormal condition? The troubleshooting strategy can be used to either return the plant to normal power operation (if troubleshooting actions correct the cause of the deviation from normal operation) or mitigate an accident condition. During troubleshooting activities, the Decision Maker uses a knowledge-based reasoning strategy to identify appropriate actions. Because the knowledge-based reasoning strategy will likely result in ordered actions, the Action Taker will transition to the "Follow Instruction" strategy.

Block 5 (Is There a Waiting Ordered Activity?): If the Decision Maker has implemented a mental procedure, an ordered activity may have been issued to the Action Taker. In order for the Action Taker to perform the ordered action, the Action Taker strategy must be "Follow Instruction". Therefore, if an ordered activity exists, the Action Taker will transition to the "Follow Instruction" strategy.

Block 6 (Follow Instruction Strategy): Activate the "Follow Instruction" strategy. This will enable the Action Taker to implement any ordered activity that has been directed by the Decision Maker.

Block 7 (Decision Maker Using Follow Procedure Strategy?): Determine if the Decision Maker has activated the "Follow Procedure" strategy. This strategy is activated by either of two possible conditions: (1) activation of the "Maintain Global Safety Margin" goal, or (2) implementation of written procedure by the Decision Maker. Because the follow procedure strategy will likely result in ordered actions, the Action Taker will transition to the "Follow Instruction" strategy.

Block 8 (Is Goal Maintain Global Safety Margin?): Is the current goal to "Maintain Global Safety Margin"? If so, the Decision Maker will implement the "Follow Procedure" strategy. In anticipation of the transition to procedure following, the Action Taker will implement the "Follow Instruction" strategy and await ordered activities.

Block 9 (Invalid Option): If the strategy selection process reaches Block 9, an error has occurred (e.g., the current goal does not match an allowed option). In this case, no strategy transition is permitted. The strategy selection process is repeated during the next decision-making processing for the operator.

Strategy Selection Process (Decision Maker)

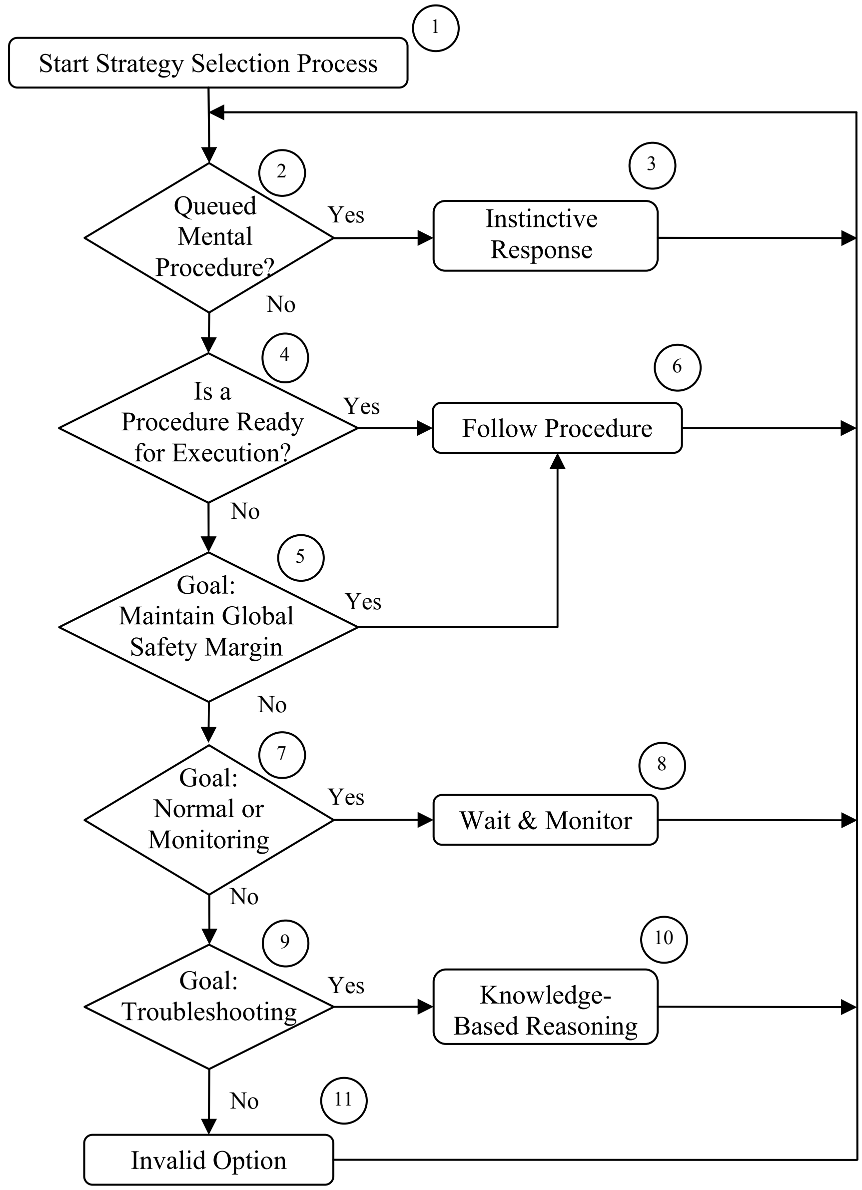


Figure 7, "Strategy Selection Process (Decision Maker)"

Strategy Selection Process (Decision Maker)

The Decision Maker can choose from three possible problem solving strategies: (1) Wait and Monitor, (2) Knowledge-Based Reasoning, and (3) Follow Procedures. The Wait and Monitor strategy is an information gathering problem solving approach intended to improve the operator's situational assessment. When using the Knowledge-Based Reasoning strategy, the operator initiates actions intended to address functional deficiencies identified through a diagnostic process. The Follow Procedure strategy involves the implementation of written procedures (e.g., abnormal or emergency operating procedures). To ensure that crew actions are coordinated, an order of precedence for problem solving strategies has been developed. The following rules guide the transition between operator problem solving strategies:

- The "Wait and Monitor" strategy has the lowest order of precedence and is only activated if no other problem solving strategy is active.

- The "Knowledge-Based Reasoning" and the "Follow Written Procedure" strategies are mutually exclusive and cannot be activated simultaneously.

- The implementation of high priority "Instinctive Response" strategy actions will interrupt all other strategies. Lower priority instinctive response actions may interrupt other strategies depending on the crew's high level goal and the individual operator profile. Once the instinctive response actions are complete, the operator will return to the previous strategy.

Block 1 (Start Strategy Selection Process): The strategy selection process executed each time the operator performs the decision-making step. The objective of strategy selection is to identify an appropriate problem solving strategy based on the operator's high level goal and preferences.

Block 2 (Is There a Queued Mental Procedure?): Check the operator's queue list of mental procedures awaiting execution. If any procedure is ready to execute, the instinctive response strategy is activated.

Block 3 (Instinctive Response Strategy): Execute the instinctive response strategy in order to follow a skill- or rule-based action. The instinctive response strategy will take precedence over all other problem solving strategies.

Block 4 (Is a Procedure Ready for Execution?): Because a memorized mental procedure can initiate the emergency operating procedures, if the Decision Maker has a queued written procedure awaiting execution, a transition "Follow Written Procedure" strategy is enabled. For example, a mental memorized procedure might cause the operator to trip the reactor and initiate the emergency procedures. This strategy transition criterion will enable the operator to initiate the emergency procedures prior to selection of the "Maintain Global Safety Margin" goal.

Block 5 (Is Goal Maintain Global Safety Margin?): Is the current goal to "Maintain Global Safety Margin"? If so, the Decision Maker will implement the "Follow Procedure" strategy to implement the written emergency operating procedures.

Block 6 (Follow Procedure Strategy): Activation of the "Follow Written Procedure" strategy enables the Decision Maker to initiate the emergency operating procedures. See the Procedure Following Flowchart for additional information.

Block 7 (Is Goal Normal Operation or Monitoring?): If the Decision Maker's high level goal is either to maintain normal operation or monitoring an abnormal condition, the Wait and Monitor problem solving strategy will be activated.

Block 8 (Wait and Monitor Strategy): Activation of the Wait and Monitor strategy enables passive information gathering (e.g., perception of alarms), active information gathering from control panel scanning, and the performance of memorized mental procedures.

Block 9 (Is Goal Troubleshooting?): Is the current high level goal to troubleshoot an abnormal condition? The troubleshooting strategy can be used to either return the plant to normal power operation (if troubleshooting actions correct the cause of the deviation from normal operation) or mitigate an accident condition. During troubleshooting activities, the Decision Maker uses a knowledge-based reasoning strategy to identify appropriate actions.

Block 10 (Knowledge Based Reasoning Strategy): Activation of the Knowledge-Based Reasoning strategy enables the Decision Maker to identify mitigative actions based on the operator's current situational assessment. Specifically, diagnoses are linked to a set of potential actions by the ***KB_ODM_Diagnosis_Actions.txt***, ***KB_ODM_Event_Matrix.txt***, and the ***KB_ODM_System_Decomposition*** input files. See the Knowledge-Based Action Execution flowchart for additional information.

Block 11 (Invalid Option): If the strategy selection process reaches Block 9, an error has occurred (e.g., the current goal does not match an allowed option). In this case, no strategy transition is permitted. The strategy selection process is repeated during the next decision-making processing for the operator.

Effect of Crew Goal on Problem Solving Strategy

Goal	Passive Information Gathering & Control Panel Scanning	Skill-Based Memorized Actions	Written Procedure Following	Knowledge-Based Problem Solving Actions
Normal Operation	☑	☑		
Monitoring	☑	☑		
Troubleshooting	☑	☑		☑
Maintain Global Safety Margin	☑	☑	☑	

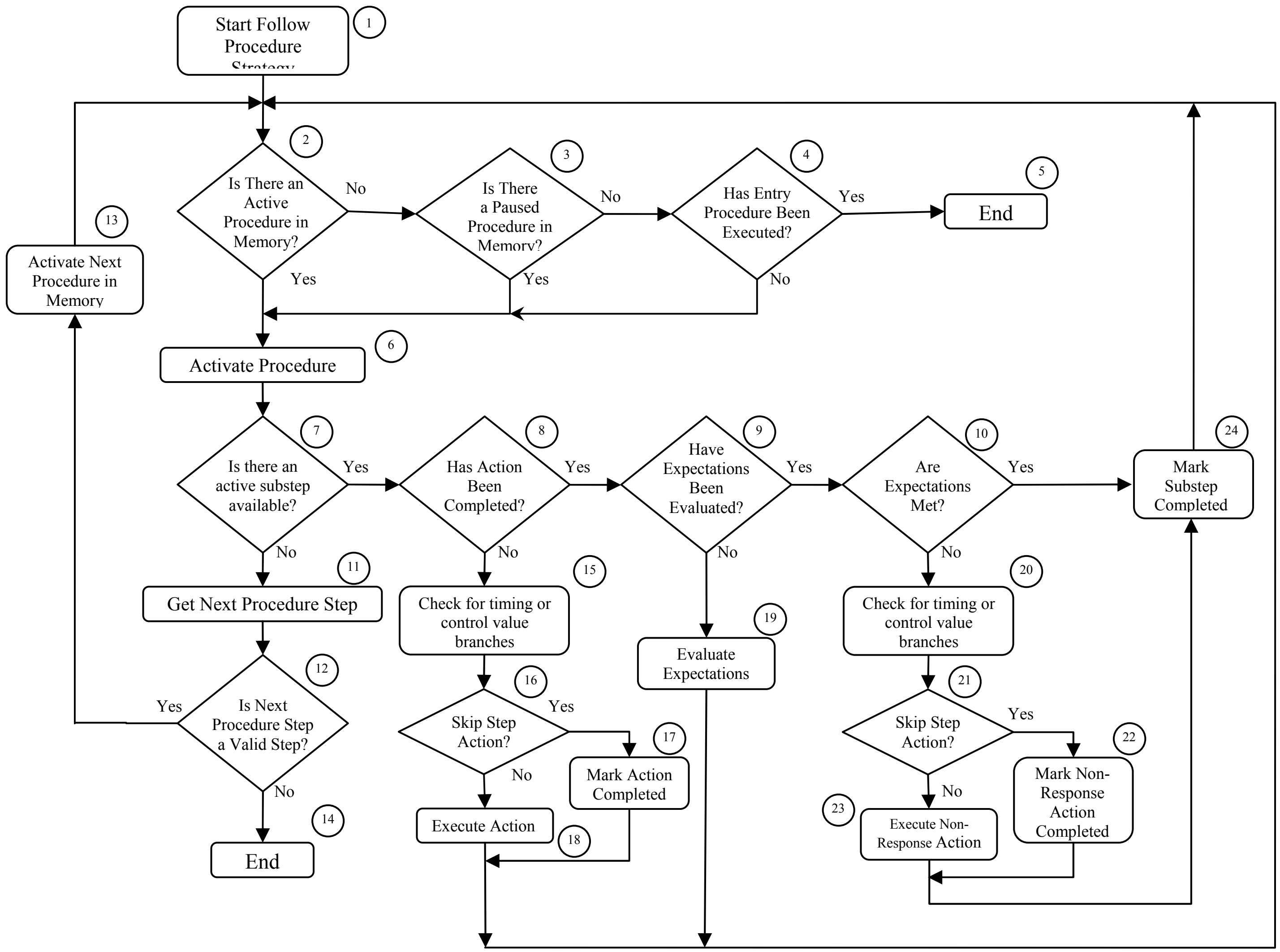


Figure 8, "Procedure Following Process"

Procedure Following Flowchart

Block 1 (Start Follow Procedure Strategy): The procedure following process is activated when the Decision Maker selects the "Follow Procedure" strategy. A similar process is also followed when any operator performs a memorized mental procedure. Procedure following is a structured problem solving strategy and it is necessary to ensure that basic rules for step sequencing are followed. Step sequencing is controlled by two separate tracking systems in ADS-IDAC: (1) a procedure status queue in each operator's memory used to track which procedure steps are active, paused, completed, or interrupted, and (2) a detailed procedure step check-off list used to execute each step element. Although each operator maintains a procedure status queue, only one detailed procedure step check-off list is kept. To prevent the crew members from interfering with each other's step sequencing, only one operator is allowed to access the detailed procedure step check-off at a time (i.e., only one operator may perform a procedure step at a time). If two operators attempt to perform the same step at the same time, the second operator (and any subsequent operators) is blocked from accessing the procedure step until the first operator has completed the procedure step and the detailed procedure step check-off list is reset.

Block 2 (Active Procedure in Memory?): Determine if the operator has an active procedure step in the memorized procedure status queue. Four categories are used to track the status of queued procedure steps:

- None (VNONE) – the procedure step has not been performed
- Active (VACTIVE) – the procedure step is in progress
- Paused (VPAUSE) – the procedure step has been temporarily suspended to permit execution of a higher priority procedure step
- Interrupted (VINTERRUPT) – the procedure step has been permanently suspended
- Completed (VDONE) – the procedure step has been completed. Note that simply completing the step does not imply that the actions have been satisfactorily executed. Equipment failures, information filtering or biasing, or the omission of step elements (step skipping) could result in the failure of a procedure step to accomplish its stated objectives.

The highest priority queued steps are those steps that are currently in progress (i.e., active steps). If the operator has more than one procedure step in an active status in their procedure queue, the last active procedure step in the queue list is selected (i.e., last in, first out queue).

Block 3 (Paused Procedure in Memory?): If there is not an active procedure step in the operator's queue list, a search is conducted for any paused (VPAUSE) procedure steps. If the operator has more than one procedure step in an paused status in their procedure queue, the last paused procedure step in the queue list is selected (i.e., last in, first out queue).

Block 4 (Has Entry Procedure Been Executed?): For plants using symptom-based procedures, the operators are generally directed to a single emergency procedure entry point (e.g., the E-0 Reactor Trip emergency procedure for a Westinghouse plant). ADS-IDAC assumes that this single procedure entry point is the first procedure step listed in the ***Procedures.txt*** input file (see Section 3). If the operator has not yet performed the entry procedure step (VNONE status), the entry procedure step will be added to the memorized queue.

Block 5 (End): If the operator's procedure queue list contains no active or paused procedures and the entry procedure step has already been performed, further procedure following activities are terminated. Terminating procedure following in this manner does not preclude reactivation of the strategy if a new procedure step is added to the operator's procedure queue list.

Block 6 (Activate Procedure): If not done already, the procedure step status is activated in the operator's queue list (i.e., the procedure status is set to VACTIVE).

Block 7 (Is There an Active Substep?): The detailed procedure step check-off list for the in progress procedure step is searched to determine if there is either an active substep or a substep that has not been performed. Each procedure substep defines a discrete action, expectation, contingency action triple. A procedure step consists of one or more substep triples and a reference to the next procedure step to be performed (if applicable). Three status tracking categories are used for procedure substeps:

- None (VNONE) – the substep has not been performed
- Active (VACTIVE) – the substep is in progress
- Completed (VSUCCEED) – the substep has been completed. Note that simply completing the step does not imply that the actions have been satisfactorily executed. Equipment failures, information filtering or biasing, or the omission of step elements (step skipping) could result in the failure of a procedure step to accomplish its stated objectives.

A substeps that either have not yet been performed (VNONE) or are in progress (VACTIVE) are considered to be active substeps. If all procedure substeps have been completed, the operator will proceed to the next procedure step.

Block 8 (Has Substep Action Been Completed?): Substep actions consist of a single interaction with the reactor plant control panel. Examples of substep actions include opening a valve, starting a pump, or actuating a control switch. If the substep had not been performed (VNONE status), the action is executed and the substep status is updated to active (VACTIVE).

Block 9 (Have Action Expectations Been Evaluated?): If the substep status is active (implying that the substep action has been completed), the status of the procedure step expectations is verified. Procedure step expectations include any parameter, alarm, mental belief, or component status verifications that are performed to determine if the substep actions had the anticipated effect. For example, after starting a pump, the operator may verify that the pump is running and that the flow rate is above a minimum threshold value. The status of expectations classified using four categories:

- None (VNONE) – the substep expectations have not been evaluated
- Active (VACTIVE) – the substep expectation verification is in progress. Since expectations can consist of multiple elements combined with Boolean operators, it may be necessary for the operator to verify more than one expectation element to completely assess the expectation status.
- Succeed (VSUCCEED) – the substep expectations have been evaluated and are met.
- Failure (VFAILURE) – the substep expectations have been evaluated and have not been met.

The substep expectations will be evaluated if the expectations have not been evaluated (VNONE) or if the expectation status is active (VACTIVE). A status of either "succeed" (VSUCCEED) or "failure" (VFIALED) indicates that the expectations have been fully evaluated. If the substep does not have any expectations, the entire substep is marked as completed.

Block 10 (Have Expectations Been Met?): If the substep expectations have been met (VSUCCEED status), or if there are no expectations associated with the substep action, the substep is updated to a completed status (VSUCCEED).

Block 11 (Get Next Procedure Step Name): If all procedure step substeps have been completed, the operator will update the status of the current procedure step as completed in the memorized queue list, reset the detailed procedure step check-off list, and obtain the next procedure step name. The next procedure step is often the next sequential step in the procedure (i.e., E-0 Step 1 references E-0 Step 2 as the next procedure step).

Block 12 (Is Next Procedure Valid?): A verification check is made to ensure that the next procedure step is a valid procedure step. In order for the step to be considered valid, it must be listed in the ***Procedures.txt*** input file and an associated ZProcedure***_procedure_name_step_name.txt*** input file must exist. If a procedure step is determined to be invalid, an error message is generated in the "***error messages.txt***" output file.

Block 13 (Activate Next Procedure in Memory): If the reference to the next procedure step is valid, the next step is added to the operator's procedure queue list and placed in an active status.

Block 14 (End): If the next procedure step is not a valid step or is identified as "NONE", the procedure following process is terminated. Terminating procedure following in this manner does not preclude reactivation of the strategy if a new procedure step is added to the operator's procedure queue list.

Block 15 (Check for Timing or Control Value Branches): Prior to performing a substep action, a check is made to determine if any timing or control value branch options have been specified for the action. These branching options are identified in the ***ProcedureActionTimeBranches.txt*** and ***ProcedureControlValueBranches.txt*** input files (see Section 3).

Block 16 (Skip Substep Action?): Prior to performing any substep action, the probability of skipping the step is dynamically calculated. If the skip step probability exceeds the threshold specified in the ***ActionTaker.txt*** or ***DecisionMaker.txt*** input files, two branches will be generated – one where the action is performed and another where the entire substep is skipped.

Block 17 (Mark Action as Completed): If a substep action is to be skipped, the action is not performed and the entire substep is marked as completed (VSUCCEED). In this case, the operator will not evaluate the substep expectations nor perform the contingency action.

Block 18 (Execute Substep Action): If the substep action is to be executed (not skipped), the action is performed.

Block 19 (Evaluate Expectations): If the substep action has been performed, a check is made to determine if the substep expectations have been evaluated (Block 9). The operator will obtain the necessary data from the reactor plant in order to determine the status of the substep expectations.

Block 20 (Check for Timing or Control Value Branches): Prior to performing a substep contingency action (also called a non-response action), a check is made to determine if any timing or control value branch options have been specified for the contingency action. These branching options are identified in the ***ProcedureActionTimeBranches.txt*** and ***ProcedureControlValueBranches.txt*** input files (see Section 3).

Block 21 (Skip Non-response Action?): Prior to performing any substep contingency action, the probability of skipping the step is dynamically calculated. If the skip step probability exceeds the threshold specified in the ***ActionTaker.txt*** or ***DecisionMaker.txt*** input files, two branches will be generated – one where the action is performed and another where the entire substep is skipped.

Block 22 (Mark Non-response Action as Complete): If a substep contingency action is to be skipped, the contingency action and the entire substep are marked as completed (VSUCCEED).

Block 23 (Execute Non-response Action): If the substep contingency action is to be executed (not skipped), the action is performed.

Block 24 (Mark Substep as Completed): At this point, the substep action has been completed (or skipped), the expectations have been evaluated (if applicable), and the contingency actions has been performed (or skipped). The all substep activities have been performed and the status is updated to completed (VSUCCEED).

Knowledge-Based Action Execution

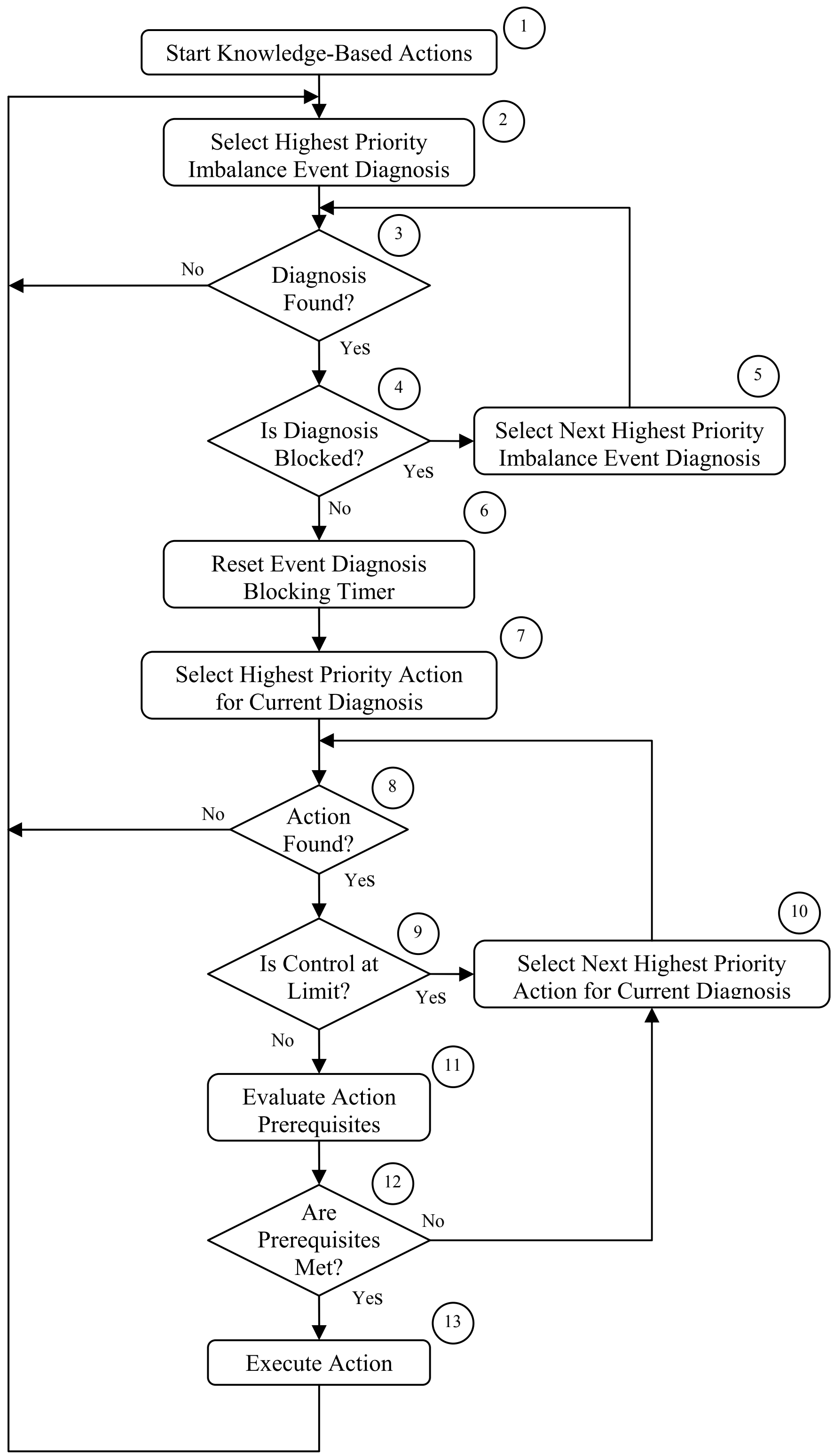


Figure 9, "Knowledge-Based Action Execution"

Knowledge-Based Action Execution

Block 1 (Start Knowledge-Based Actions): The knowledge-based action execution process is activated when the Decision Maker selects the "Troubleshooting" problem solving strategy. The overall objective of the knowledge-based action execution process is to select appropriate actions based on the operator's situational assessment. The situational assessment is a diagnostic process intended to identify functional degradations in reactor plant thermal hydraulic processes. For example, during a loss of coolant accident, an operator may perceive functional degradations associated with maintenance of reactor coolant inventory and core energy removal. These functional degradations are called "Imbalance Events" because they generally are associated with mass, energy, or momentum imbalances in the main reactor plant systems. The ***KB_ODM_Diagnosis_Actions.txt*** links each imbalance event to a set of potential actions. The operator selects an appropriate action based on two factors: (1) the relative priority of the action and (2) the status of the prerequisites associated with the action.

Block 2 (Select Highest Priority Imbalance Event Diagnosis): A search of all diagnosis events categorized as "Imbalance" events listed in the ***KB_ODM_Event_Matrix.txt*** is conducted to identify the imbalance event with the highest confidence level. The highest priority event is selected as the primary driver for selecting a knowledge-based action.

Block 3 (Diagnosis Found?): The confidence level of the highest priority imbalance event is must exceed the minimum cutoff level set in the DaignosisActionUnit.cpp class definition[3]. The default value for the minimum confidence level is 0.5. If the highest priority imbalance event diagnosis does not exceed this value, no knowledge based actions will be performed and the knowledge-based action execution process will be terminated. Terminating the knowledge-based action process in this manner does not preclude future performance of a knowledge-based action should new information increase the confidence level in an imbalance diagnosis above the minimum threshold value.

Block 4 (Is Diagnosis Blocked?): Each imbalance diagnosis listed in the ***KB_ODM_Diagnosis_Actions.txt*** input file includes a reset delay time. The reset delay time is intended to prevent the highest priority imbalance diagnosis from monopolizing the operator's attention and preventing the operators from addressing lower priority events. After the operator selects an imbalance event action group, future re-activation of the same imbalance event is blocked until the reset time expires. By adjusting the length of the reset delay time, the analyst can control the distribution of crew resources to higher and lower priority imbalance events. If the reset delay time is inhibiting the highest

[3] The minimum confidence level threshold can be changed within the ADS-IDAC program by revising the value of the "highest_confidence" variable in the DiagnosisActionModule::getPtrToHighPriorityDiagnosisActionUnit() class description. The default value of this variable is currently set to 0.5. This variable will be moved to an appropriate input file during a future ADS-IDAC revision.

priority imbalance diagnosis, the process returns to Block 3 to find the next highest priority diagnosis. This process repeats until either: (1) the highest priority unblocked imbalance event is found, or (2) the confidence level of no unblocked imbalance event exceeds the minimum threshold in Block 3.

Block 5 (Select Next Highest Priority Diagnosis): If the highest priority diagnosis is blocked because the reset time has not expired, the operator will attempt to select the next highest priority diagnosis.

Block 6 (Reset Event Diagnosis Blocking Timer): The reset delay timer for the selected high priority imbalance diagnosis is reset. The imbalance diagnosis is blocked from further activation until the reset timer expires.

Block 7 (Select Highest Priority Action for Current Diagnosis): Each imbalance event can be associated with one or more possible actions (see the ***KB_ODM_Diagnosis_Actions.txt*** input file description in Section 3). Actions are assigned a priority level and may be associated with one or more prerequisites. During this step, all actions associated with the imbalance event are reviewed and the action with the highest priority is selected. If two or more actions are tied for the highest priority level, the action listed first in the ***KB_ODM_Diagnosis_Actions.txt*** is selected.

Block 8 (Action Found?): If an imbalance event is not associated with any knowledge-based actions no action for the imbalance event can be executed and further knowledge-based action activities are terminated. However, if during the next ADS-DAC time step, the reset time delay for this imbalance event allows a different high priority imbalance event diagnosis to be identified, an appropriate knowledge-based action might be found.

Block 9 (Is Control at Limit?): Some actions may be associated with physical limits that either preclude execution of the action or make execution of the action unnecessary. For example, once a control valve is fully opened, executing an action to open the valve would have no impact on the reactor plant. Similarly, once the reactor is tripped, executing an action to trip the reactor again is redundant and not necessary. Therefore, knowledge based actions are checked to verify that the action is capable of changing the state of the reactor plant. If the action would not result in a change in state (e.g., the action had already been implemented), the action is at the control limit and will not be executed. In this case, an alternate (and potentially lower priority action) would be selected in block 10.

Block 10 (Select Next Highest Priority Action): This block is entered if: (1) execution of the action would not change the reactor plant state, or (2) the associated prerequisite conditions are not met. In these cases, an alternate (and perhaps lower priority action) is selected for further evaluation.

Block 11 (Evaluate Action Prerequisites): Each knowledge-based action can be associated with a set of prerequisites conditions. All prerequisites must be met before the action can be performed.

Block 12 (Are Prerequisites Met?): If the specified action prerequisites are not met, the associated action cannot be performed. An alternate (and perhaps lower priority) action will be selected in Block 10.

Block 13 (Execute Action): The selected knowledge-based action is executed.

1. Input File Format

ADS-IDAC requires the user to provide an extensive amount of information related to reactor plant systems, operator knowledge and skills, and crew preferences and tendencies. The ADS-IDAC user provides this information in a collection of input files. Section 3, "Input File Format," of this manual provides a detailed discussion of the input files required to run the ADS-IDAC code. For each input file, a discussion of the purpose, file format, description of input options, and an input file sample are provided. In order to standardize the discussion of the input files, the following conventions are used:

- *variable types:* In general, input data can be any of three possible data types: double, integer, or string. "Double" refers to real valued numbers (e.g., 3.14), "integer" refers to an integer number (e.g., 1, 0, -12), and "string" refers to a word comprised of character elements (e.g., "Reactor_Trip"). Although the current version of ADS-IDAC contains some error checking mechanisms, the

- ***bold-italicized*** *words:* Generally, input variables that must be provided by the analyst are denoted by the use of ***bold italics***. The detailed input description for each input file provides the data type (integer, double, string), the allowable data range, and other data options.

- *Quotation Marks (" "):* Quotation marks are used to identify input file names (e.g., "ControlPanel.txt").

- *conditional input**:*** Certain input data quantities depend on the specific set of options selected by the code user. For example, ADS-IDAC allows the control room crew to use several different methods to interact with reactor plant components (see General Note 5, "Operator Control Inputs"). Depending on the selected control option, the data input requirements can change. For clarity, these conditional inputs are enclosed in brackets ("[ ]"), and should only be provided when the associated input option has been selected.

- *multiple data entries:* Often, an input file can include multiple entries for similar data types. For example, the input file that describes the reactor plant control panel (ControlPanel.txt) allows the user to enter multiple alarms, controls, and indicators. However, the input file descriptions in Section 3 typically provide only a representative input example. Where multiple data entries are permitted, the three lines following the representative example are annotated with only a dot ("**.**") to indicate the option for multiple data entries.

2. Input Strings

When string variable are entered in ADS-IDAC input text files, they must be entered without any spaces. The underbar character (_) may be used when it is necessary to separate words. For example, the string ***Reactor Trip*** should be entered as ***Reactor_Trip***. Because ADS-IDAC expects the input files to follow a consistent formatting convention, this convention prevents a two word string from being misinterpreted as two separate input quantities. The one exception to this rule is if the string is enclosed with quotation marks. The detailed file format descriptions provided in this manual describe the allowable string format for each input file.

3. Integer Codes

In order to make certain integer variable values more meaningful within the ADS-IDAC coding, a catalog of integer codes has been developed. These codes cover commonly used integer variable values and are often easier to work with than string values. For example, the integer variable that stores the current operator problem solving strategy can be set to represent any of several possible strategies such as “Follow Written Procedure”, “Knowledge-Based Reasoning”, or “Wait and Monitor”. Because use of an integer value is more efficient from both a program execution speed and memory storage perspective, each of the possible strategies is assigned a unique integer code number. To make the ADS-IDAC source code more readable, the assigned code numbers are stored in a series of variables described in the ***TermConversion.h*** C++ header file (located within the ADSSource directory). To illustrate this concept, the following codes are currently used for the operator problem solving strategy:

Problem Solving Strategy	Variable Name	Variable Integer Value
Wait and Monitor	VSWM	3054
Follow Instruction	VSFI	3062
Knowledge-Based Reasoning	VSIDR[4]	3058
Follow Written Procedure	VSFP	3059

In general, the name for all coded variables begins with the letter “V”. The remainder of the variable name is either a descriptive noun name or acronym for the quantity being represented. Using this coding scheme, the ADS-IDAC programmer can work with the more descriptive variable name while still gaining the computational advantages of using a single integer quantity to represent a string value. Unfortunately, several input files currently require the user to enter the variable integer value for a quantity rather than the more descriptive variable

[4] The Knowledge-Based Reasoning had previously been called “Inductive/Deductive Reasoning”. Therefore, the variable name VSIDR has been used for this strategy.

name. Consequently, the user needs to be aware of this coding scheme. When this convention is used in an input file, the input description identifies the allowable integer values.

A listing of integer codes used in ADS-IDAC is provided in Section 5, "Term Conversions".

4. Procedures

Two main types of procedures are used in ADS-IDAC: (1) written procedures, and (2) memorized mental procedures. Written procedures represent formal proceduralized guidance contained in normal, abnormal, and emergency operating procedures. Memorized mental procedures represent the skill- and rule-based actions routinely used by the operators that do not require formal procedure guidance. The filename for all ADS-IDAC procedure input files use the prefix "***ZProcedure_***". Mental procedures are distinguished from written procedures by the prefix "***ZProcedure_MPBG_***"[5].

Generally, a written procedure is continued until the procedure is completed. However, the procedure flow may be interrupted by procedure transfers (which direct the crew to a different procedure), activation of a mental belief that activates a memorized mental procedure, or abandonment of the "Follow Written Procedure" strategy. Two types of procedure transfers can be modeled: (1) a permanent procedure transfer and (2) a temporary transfer to an auxiliary procedure followed by resumption of the initial procedure. An example of a permanent procedure transfer is the transfer from a general reactor trip procedure to a more specific emergency procedure (e.g., transfer from the Westinghouse E-0 to E-3 procedures during a steam generator tube rupture event). A temporary procedure transfer is used when the crew temporarily interrupts an active written procedure to follow a functional recovery guideline to address a degraded condition. When a permanent transfer is executed, the original procedure is terminated and will not be reactivated when the new procedure is completed. When a temporary transfer is executed, the original procedure will be recommenced at the step where it was interrupted when the new procedure is completed. ADS-IDAC executes a temporary procedure transfer when the new procedure is either a mental procedure or has the prefix "***ZProcedure_FRG_***" (indicating the new procedure represents a functional recovery guideline).

For additional flexibility, the crew may be directed to transition to a written procedure from a memorized mental procedure. This may be the case following a manual reactor trip, where the crew scrams the reactor and initiates a general reactor trip emergency procedure (such as E-0). Because only the Decision Maker is permitted to direct the performance of a written procedure, ADS-IDAC

[5] The acronym "MPBG" for mental procedures can be interpreted as "Mental Procedure – Belief Generated". These procedures are generally initiated by the activation of a mental belief contained in the ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** input file.

places restrictions on the types of procedure transitions available to each operator. The Action Taker may initiate a memorized mental procedure and transition to other mental procedures, but may not initiate or transition to a written procedure. The Decision Maker may initiate and transition between all procedure types.

5. Operator Control Inputs

ADS-IDAC provides four possible control inputs for each component that can be operated by the control room crew:

1) changing the component operating mode (e.g., automatic vs. manual mode),
2) setting a specific control value for a component (e.g., throttling control valve to 50% open),
3) incrementing the control setting of a component (e.g., throttling open a control valve by an additional 10%), and
4) setting a control value based on a perceived parameter (e.g., setting the steam dump target pressure equal to the perceived main steam header pressure).

These capabilities provide sufficient flexibility to realistically model all significant operator interactions with the plant model. The details of implementing each of these control inputs are provided in Section 3 of this manual.

6. Input File Consistency

Certain ADS-IDAC input files contain cross references to items described in other input files. For example, an input file for a procedure step may contain a cross reference to an indicator or a control described in the control panel input file. To avoid errors during input file processing or program execution, it is important to ensure that any cross referenced data in the input files are internally consistent. The input file formatting guidance included in Section 3 highlights areas where data is cross referenced across multiple input files. Prior to running the ADS-IDAC, all input files should be reviewed to verify the consistency of data references. To facilitate this consistency review, the following input data files should be checked:

<u>Control Panel Information</u>

- ☐ Control panel indicators, controls, and alarms included in the ***ControlPanel.txt*** input file should refer to valid ***control_volume_names***, ***control_junction_names***, ***variable_trip_names***, ***logical_trip_names***, ***interactive_control_names***, and ***control_variable_names*** in the ***RELAP5_channels.txt*** input file

☐ References to plant controls (e.g., valves, pumps, actuators) contained in the following input files should match a valid ***control_name*** in the ControlPanel.txt input file:

- procedure step descriptions (***ZProcedure_procedure_name_step_name.txt***)
- control value branching options (***ProcedureControlValueBranches.txt***)
- action time branching options (***ProcedureActionTimeBranches.txt***)
- conditional component failure events (***SystemReliability.txt***)
- knowledge-based actions (***KB_OAT(ODM)_Diagnosis_Actions.txt***)
- system decomposition (***KB_OAT(ODM)_System_Decomposition***)
- initiating events (***Initiating_Event.txt***)

☐ References to plant parameters, component states, and alarms contained in the following input files should match a valid ***parameter_name***, ***component_name***, or ***alarm_name*** in the ControlPanel.txt input file:

- procedure step descriptions (***ZProcedure_procedure_name_step_name.txt***)
- action time branching options (***ProcedureActionTimeBranches.txt***)
- conditional component failure events (***SystemReliability.txt***)
- knowledge-based action prerequisites (***KB_OAT(ODM)_Diagnosis_Actions.txt***)
- system decomposition (***KB_OAT(ODM)_System_Decomposition***)
- mental belief prerequisites (***KB_OAT(ODM)_HardWired_Diagnosis.txt***)
- failed or biased instruments (***KB_OAT(ODM)_Bias_Factors.txt***)
- periodically scanned indicators (***KB_OAT(ODM)_Scanned_Parameters.txt***)
- time constrained parameters (***KB_OAT(ODM)_Time_Constrained_Input.txt***)
- critical safety parameters (***KB_OAT(ODM)_Safety_Parameters.txt***)

<u>Operator Knowledge Base</u>

☐ Key plant functions described in the operator system decomposition input file (***KB_OAT(ODM)_System_Decomposition***) should be included in the event diagnosis matrix (***KB_OAT(ODM)_Event_Matrix.txt***).

☐ Each symptom referenced in the event diagnosis matrix (***KB_OAT(ODM)_Event_Matrix.txt***) should be associated with a mental belief in the ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** input file.

□ Any mental belief which is used as a prerequisite condition for another mental belief should be included in the ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** input file.

Procedures

□ All procedure steps listed in the ***Procedures.txt*** should be associated with a unique procedure step input file (***ZProcedure_procedure_name_step_name.txt***)

□ Procedure names references in mental beliefs (***KB_OAT(ODM)_HardWired_Diagnosis.txt***) should refer to a valid procedure name (***ZProcedure_procedure_name_step_name.txt***)

□ The following branching option files should refer to valid procedure names (***ZProcedure_procedure_name_step_name.txt***):

- control value branching options (***ProcedureControlValueBranches.txt***)
- action time branching options (***ProcedureActionTimeBranches.txt***)

□ Mental procedure branching options specified in the ***MentalProcedureActivationTimeBranches.txt*** input file must reference a valid mental belief in the ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** input file.

The failure to ensure consistency between input files may result in unusual or unexpected errors during input processing or program execution. Although ADS-IDAC contains many input data checking features, the user should not rely on these program checks to ensure validity of the input data files.

7. Branch Generation

Some procedure actions require the operator to interact with the reactor plant or coordinate with other crew members. Typical examples of these tasks include checking the status of plant equipment, manipulating a control, or verifying a parameter value. Because these tasks require some amount of time to complete, further operator decision-making process is suspended until the task is complete. Within the ADS-IDAC, the generation of these tasks is accomplished by creating a new branch event. A branch represents a discrete event or information chunk created during a simulation sequence. Branch generation in this context should be distinguished from the creation of branches in a discrete dynamic event tree (DDET). Most branches do not cause multiple accident sequences to be generated. However, when two or more branches of the same type are generated in a single time step, ADS-IDAC splits the current sequence path into two or more separate paths. The splitting of a sequence into two or more sequence paths corresponds to the generation of a DDET branching event. Branching rules are used to generate DDET branching events during an ADS-IDAC simulation.

8. Controlling Branch Generation With Branching Rules

Branching rules can be constructed to reflect variations in plant hardware and crew responses to plant events. ADS-IDAC provides the capability to generate branching points for the following system and operator performance attributes during a simulation:

- Use of memorized information – Use of this branching rule generates two branches: one where the operator will use previously perceived information (if available) and another where the operator will always obtain recent information from the control panel. The use of memorized information can increase the action execution speed of the operator but may result in the use of outdated information.
- Troubleshooting probability – Use of this branching rule generates two branches: one where the operator will attempt to use knowledge-based actions to mitigate the accident event and another where the operator will immediately implement the emergency operating procedures when an abnormal condition is detected. Use of knowledge-based actions can result in more efficient accident mitigation but might result in inappropriate actions if the operator's situational assessment is incorrect.
- Procedure use probability – This branching rule provides a simulation control function to switch the operator from a knowledge-based mitigation strategy (if enabled) to the procedure following strategy.
- Mental belief branch probability – Use of this branching rule generates two branches when the necessary prerequisite conditions for a mental belief are met: one where the mental belief is activated, memorized, and used to implement the associated mental procedure and a second branch where the mental belief remains inactivated.
- Mental procedure activation time – Following activation of a mental belief, the operator will implement the associated mental procedure (if one is specified) after the activation time delay has elapsed. To model the uncertainty associated with this parameter, the mental procedure activation time is represented by a three parameter Weibull probability distribution. This branching rule allows the analyst to generate two or more branches, each of which represents a sample taken over a different partition of the activation time probability distribution.
- Equipment failure and recovery – The current ADS-IDAC component reliability module models only demand failures (e.g., failure to start). This branching rule generates failure and success branches when component operation is demanded. If the operator attempts to restart failed equipment, branches representing component recovery and permanent failure are

generated. Thus, each equipment failure event can result in three outcomes: (1) the equipment does not fail, (2) the equipment initially fails but is later recovered, and (3) the equipment fails and is unrecoverable.

- Procedure step skipping – When an operator executes either a mental or written procedure step, either the step actions or contingency actions can be skipped. When stepping skip is activated, two branches are generated when procedure actions are executed: one branch where the action is performed and another branch where the action is omitted.
- Action control value branches – For procedure step actions associated with a quantitative control input (e.g., opening a throttle valve to 10% open), two or more branches can be generated to explore the effect of variations in the control input.
- Action time branches – Every operator action is associated with a specific action execution time. To model the uncertainty associated with this parameter, the action time is represented by a three parameter Weibull probability distribution. This branching rule allows the analyst to generate two or more branches, each of which represents a sample taken over a different partition of the action time probability distribution.

By appropriately combining these branching rules, a wide spectrum of possible plant and operator states can be simulated.

9. Terminating Sequences

In order to achieve complete scenario coverage using dynamic probabilistic risk assessment methods, it is necessary to explore a large number of accident sequences. However, simulating a large number of sequences often requires a significant amount of computational power and time. Therefore, uninteresting sequences are often terminated or truncated. ADS-IDAC provides several methods to terminate or truncate accident sequences. Truncation methods are based on sequence elapsed time, sequence probability, or conditional events. The following truncation methods are available in ADS-IDAC:

- *Sequence Truncation Time*: This option allows the user to set a maximum simulation time limit for each sequence. The time limit is set in the ***ZCtrlPar.txt*** input file. It is important that the RELAP time control cards (Card2 200-299) set a maximum simulation time at equal to or greater than the sequence truncation time (or the RELAP code will terminate the sequence early)

- *Sequence Probability*: This option terminates a sequence when the sequence probability is less than the cutoff value specific in the ***ZCtrlPar.txt*** input file.

- *Procedure Non Response Exit*: This option allows the user to terminate a sequence if a set of procedure step expectations are not met. This option is activated by specifying the keyword “VSTOP” in the ***non_response_action*** type field in the appropriate **ZProcedure*_procedure_name_step_name*.txt** input file. When this option is used, if the operator determines that the associated procedure step expectations are not met, the sequence will terminate.

- *Alarm Activation*: This option terminates the sequence if an alarm with the prefix “A_ENDSEQ” is actuated. The user specifies the alarm name and the actuation conditions in the ***ControlPanel.txt*** input file

10. Running ADS-IDAC

The ADS-IDAC code can be run from either a compiled executable file or from the Microsoft Visual C++ Developer Studio[6]. In either case, the ***ZiniADS.txt*** must be located in the same directory as either the executable version of ADS-IDAC or the main project file for the Fortran and C++ code. The ***ZiniADS.txt*** file provides the location of RELAP thermal hydraulic plant model input deck and the ADS-IDAC input files.

Units

ADS-IDAC can be run using either SI units or British Units. Changes between the base units must be made from within the Microsoft Visual C++ Developer Studio. There are currently no provisions to change base units from the input files. In order to toggle between SI and British units, the following line must be revised in the ***schedule.cpp*** class file:

R5PAR_mp_SI_UNIT = false;

If R5PAR_mp_SI_UNIT is set to “false”, the base unit is set to British (e.g., psi, F, lbm/sec). If R5PAR_mp_SI_UNIT is set to true, the base unit is SI (e.g., Pa, K, kg/sec).

Time Step

There are two time steps of interest to the user when running ADS-IDAC: the RELAP thermal hydraulic model time step and the ADS-IDAC time step. The RELAP time step is set by lines 200-299 in the RELAP input deck. To avoid early termination of a simulation run, the maximum RELAP time should be set to a value greater than the truncation time set in ***ZCtrlPar.txt***. The ADS-IDAC time step is set to a default value of 0.5 seconds, but can be adjusted manually from within the Microsoft Visual C++ Developer environment. In order to adjust the ADS-IDAC time step, the following changes must be made:

---

[6] ADS-IDAC is written in two main programming languages – Fortran and C++. The RELPA thermal hydraulic code and related interface infrastructure is written in Fortran while the remainder of the ADS-IDAC code is written in C++.

- Modify Fortran module ***dtstep.f*** by changing the ***ADS-IDAC_time_step*** to the desired value (this code line is located at approximately line 1505). The default value for ***ADS-IDAC_time_step*** is 0.5 seconds.

  if((timehy - timesno) .ge. ***ADS-IDAC_time_step****) then*

- Modify C++ module ***CommandControlCenter.cpp*** to the same time step value set in ***dtstep.f***:

  *this->reduceRemainingTime(****ADS-IDAC_time_step****)*

These changes ensure that the ADS-IDAC scheduler time step remains synchronized with the RELAP5 thermal hydraulic model.

Error Log File

ADS-IDAC generates an error log file if any error or warning is generated during a simulation run. Following a simulation run, the user should determine if the error log file ***err messages.txt*** was placed in the ADS-IDAC main directory. Some errors will cause termination of the simulation, while others are less severe and simply alert the user to an unexpected condition.

Section 2: Modeling Guidance

1. Modeling Crew-to-Crew Variability

Even when subject to similar personnel selection, training, and administrative requirements, nuclear plant control room operators can exhibit significant crew-to-crew variability during non-routine events. A strength of the ADS-IDAC simulation approach is the ability to systematically model the sources of variability. In general, ADS-IDAC captures crew variability within three main categories: procedure adherence and implementation, crew knowledge and experience, and individual preferences and tendencies. ADS-IDAC allows the analyst to create branching events to address each of these areas of variability.

2. General Guidance

The initial step in addressing operator variability is to gather information pertaining to crew behavior. The following information should be collected:

- Written Procedures: All procedures associated with the event(s) to be analyzed should be collected. In addition procedures associated with plant operation, administrative procedures that guide crew decision-making should also be reviewed. Typically, plant administrative procedures will address factors such as communication, the general conduct of operations, self-checking behavior, activities considered to “skill-of-the-craft” that do not require written instructions, and guidelines for procedural adherence.

- Training Materials: Operators form mental models about plant behavior as a result of their training and experience. Training materials, including lesson plans, simplified plant drawings, and simulator training scenarios should be reviewed to identify key mental models and diagnostic processes employed by the operators. In particular, training materials might include “rules of thumb” or other non-proceduralized shortcuts that operators may use as an aid to operation.

- Operating Experience: Review plant operating experience associated with the event(s) to be analyzed to identify actions taken by actual control room crews. Identify instances when crews have performed unexpected or non-proceduralized actions or developed a mental model of the plant that differed from the actual plant conditions. Potential sources of operating experience include NRC inspection reports, generic communications, licensee event reports, and NUREG reports.

- Expert Elicitation: Consult operations experts (e.g., training instructors, inspectors) to identify areas where crews might have difficulty in responding to an accident event.

3. Specific Guidance

i. Identify potential sources of crew-to-crew variability. Review operations procedures, operator training, and crew preferences and tendencies using the following guidelines:

a) Operating Procedures

1) Review procedures and identify major control actuations (stop/start significant pumps, open/shut major valves, control manipulations (initiate/stop depressurizations, cool downs).

- These actions often delineate major phases in the emergency procedures – crews may hold a briefing prior to initiating significant plant actions. The conduct of briefings and meetings can be addressed by the inclusion of procedure hold points with action time branching rules.

- Identify possible operator control manipulation variations. In particular, focus on control manipulations where the operators may use too little or too much control input (e.g., opening valve more or less than desired). Variations in control inputs can be modeled with appropriate control value branching events.

2) Identify ambiguous criteria in procedure steps by searching for key words such as "increasing," "steady," "decreasing," and "stable". These keywords highlight steps where crews may exhibit variability in interpreting procedural criteria. For example, crews may use different threshold criteria for judging acceptability. Further, the criteria may depend on the situational context and depend on the crew's knowledge and experience. For example, a crew might conclude that a decreasing trend in pressure actually indicates a stable condition if the cause of the decrease is well understood and a direct result of operator actions. Crew variability associated with interpretation of potentially ambiguous criteria can be modeled with several conditional simulation runs to explore a range of threshold values.

3) Assess the main objective of each procedure step. While some procedure steps may support the main procedure objectives, other steps may be associated with supporting activities that may have a lower salience or priority for the operators. For example, actuation of safety injection following a loss of coolant accident would be expected to have a greater relevance to the assessed situation than shutdown of a feedwater heater. Therefore, steps that are more directly linked to the main procedure objectives would be expected to have a lower likelihood of being skipped. However, some steps that do not appear to be directly linked to the procedure objectives may improve the efficiency of the procedure, establish

prerequisite conditions that support later actions, or delay the onset of core damage. Examples of such actions include:

- Tripping the reactor coolant pumps following a loss of secondary heat sink. This action reduces heat input to the reactor coolant system but does not directly support the main procedure objective to recover feedwater flow to the steam generators. Consequently, the action has a safety impact (reduced time available for recovery) but may not be salient to the operators.

- Blocking automatic safety injection actuation signals prior to an intentional cooldown or depressurization to prevent an unnecessary safety system actuation. The failure to block an actuation signal under these circumstances can unnecessarily delay mitigative actions, but may not have a high degree of relevance to the operators during the accident.

The analyst can categorize procedure step objectives by setting the ***step_type*** variable in the ***ZProcedure_procedure_name_step_name.txt*** input file (see Section 3).

4) Identify all steps that require that require continuous monitoring by the operator such as continuous action steps or fold out page actions. These steps often require the operator to monitor the state of a targeted plant parameter and initiate an action or procedure transfer if a critical threshold is passed. If the operator fails to monitor the target parameter at regular intervals, initiation of the required actions could be delayed. The operator's tendency to periodically monitor the target parameter is established in the ***ActionTaker.txt***, ***DecisionMaker.txt***, and the ***KB_OAT(ODM)_Scanned_Paramter.txt*** input files. Performance variabilities in the initiation of actions directed by continuous action steps or fold out page steps can be captured with appropriate mental beliefs in the ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** input file.

5) Identify procedure steps that refer to components that have failed during the accident scenario. These steps may trigger the operator to initiate recovery or compensatory actions. These actions can be modeled with appropriate mental beliefs in the ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** input file.

b) Mental Beliefs and Memorized Actions

1) Based on operator training and experience, identify mental beliefs that the crew is likely to reach during the scenario. Examples might include "uncontrolled steam generator level increase" for a tube rupture scenario or "faulted steam generator" for a main steam line break. These mental beliefs represent the crew's situational assessment and their underlying mental

models. Crew variability might arise from the failure to reach appropriate belief states, delays in acting on beliefs, or setting a higher or lower evidence threshold for a belief. These factors can be included in the ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** input file

2) For the specific scenario, identify actions that the crew might take to increase the time available until an automatic safety feature is actuated. Examples include maximizing reactor coolant system makeup during a primary system leak or reducing steam loads during a partial loss of feedwater. These actions often represent non-proceduralized actions arising from the operator's training and experience. These actions can be activated by appropriate mental beliefs in the modeled by with mental beliefs in the ***KB_OAT(ODM)_HardWired_Diagnosis.txt*** input file and described in by a mental procedure step (***ZProcedure_MPBG_procedure_name_step_name.txt***)

3) Identify conditions under which crews might manually activate safety systems. These conditions can be captured by mental beliefs which lead the crew to perform the manual action. For example, a control room might initiate a safety injection when pressurizer level is low.

4) Identify potential control system failures that provide symptoms similar to the actual accident condition. Operators could potentially confuse an accident event with a minor control system failure and activities to address the misdiagnosed failure may distract the crew from addressing the accident. The ***KB_OAT(ODM)_Event_Matrix.txt*** input file should include a wide spectrum of simple failures, abnormal conditions, and emergency events to more realistically represent the crew's diagnostic behavior.

c) <u>Crew Preferences and Tendencies</u>

1) Assess the tendency of crews to rely on previously memorized information rather than use of recent information obtained directly from the control panel indicators. The use of memorized information can reduce activity execution time but may result in the use of outdated and incorrect information. The crew tendency toward the use of memorized information can be set in the ***ActionTaker.txt*** and the ***DecisonMaker.txt*** input files.

2) Assess the crew tendency to procedural adherence. Procedural steps in ADS-IDAC have three main components: (1) initial action activity, (2) expectations associated with the initial action activity, and (3) a non-response action that is executed if the action expectations are not met. The operator may skip either the initial action activity or the non-response action (evaluation of the action expectations cannot be skipped). The probability of skipping the initial action or non-response action is dynamically calculated based upon the baseline skip probability for the step component

(specified in the procedure step input file), the type of procedure being followed, the step objectives, the relevance of the action to the operator's situational assessment, and certain PIFs. If the calculated skip probability for the initial action activity exceeds a preset threshold, the procedure step is skipped by the crew (an alternate branch is also generated where the step is performed). The skip step threshold can be set in the ***ActionTaker.txt*** and the ***DecisonMaker.txt*** input files to represent a crew's tendency to follow written procedures.

3) Determine the crew's threshold level for concluding that an accident has occurred. Based on accumulated evidence, the ADS-IDAC diagnosis engine calculates a value representing the potential for the observed information being related to an accident condition. When this calculated value exceeds a preset threshold, the crew will transition to the emergency operating procedures and shutdown the reactor. Crew's may exhibit variability in their tendency to delay initiation of the emergency procedures (perhaps to allow other mitigative actions to work) or require strong evidence prior to initiating a shutdown. This variability can be captured in the ***ActionTaker.txt*** and the ***DecisonMaker.txt*** input files.

4) Tendency to rely on knowledge-based troubleshooting rather than written procedures. Operating crews may attempt to address an emergency condition through the use of actions based on their knowledge and experience rather than through written procedures. ADS-IDAC includes options in the ***ActionTaker.txt*** and the ***DecisonMaker.txt*** input files to allow the analyst to specify the crew's tendency to use knowledge-based reasoning approaches to problem solving rather than written procedures.

5) Identify factors that influence the procedure execution speed for the crews. Depending on certain organizational factors, familiarity with the procedures, and experience crews might have significant variability in their procedure execution speed. Variations in execution speed can be modeled in ADS-IDAC with the use of procedure hold points. Potential hold points include the start of a new procedure (including transfers between procedures) or prior to accomplishment of a major procedural action (such as cooldown, depressurization, initiation of safety injection, etc). The analyst can introduce a "dummy" procedure step to provide a suitable time delay to accommodate variations in execution speed. Branching options can also be exercised to simulate fast, slow, or nominal crews.

Table 1, "Mapping Crew Variability to ADS-IDAC Branching Rules," summarizes the potential sources of crew variability and the associated ADS-IDAC branching rules.

ii. List all branching events that are applicable to analysis. Branching events include the sources of crew-to-crew variability identified in the previous step and hardware related events. Hardware related events include initial failures, conditional failures, and potential operator recovery actions.

iii. Define the specific parameter values that will be used to model branching events. Parameter values include:

- Branching Probabilities – In general, the Beta Distribution is used to describe hardware failure and recovery probabilities. Branching events associated with mental belief activation use point probability estimates. The analysts is required to supply either the Beta Distribution parameter estimates (α and β) for hardware failures or a point estimate for mental belief activation probabilities.

- Timing Parameters – The time required to perform procedure steps and certain decision-making activities are modeled with a three parameter Weibull distribution. The time required to exchange crew communications and transition between procedures is modeled with a point estimate.

- Control Value Variability – Identify the control values and probability point estimates that will be used to model variations in control inputs

- Timing Variability – Identify the procedure steps where action timing branches should be generated. The number of branches that will be generated should also be determined.

- Mental Belief Activation Time Variability – Identify the mental beliefs and number of associated branching events that will be used to model variabilities in initiating skill- and rule-based actions arising from mental beliefs.

- Goal and Problem Solving Strategy Tendencies – Identify the crew tendencies for goal and problem solving selection. These tendencies are represented by point probability estimates in the ***ActionTaker.txt*** and ***DecisonMaker.txt*** input files.

A detailed description of the required parameter values is provided in Section 3. Table 2 provides a suggested format for summarizing the branching events that will be used for the analysis.

iv. Define sequence end states. There are two methods for terminating a sequence when a critical parameter exceeds a threshold value: (1) a "VSTOP" non-response procedure action, or (2) activation of a end sequence alarm (i.e., an alarm with a "A_ENDSEQ_" prefix). The analyst should identify the critical parameters and thresholds that are used to terminating a sequence. For example, a fuel uncovery condition indicated by a low reactor vessel water level could be indicative of the onset of core damage.

v. Develop a simulation matrix that defines the detailed computer simulations that will be performed. Due to the computer processing limitations, it is not currently practicable to run a single ADS-IDAC simulation capable of exercising all branching points. The main difficulty is the exponential increase in accident sequences as the number of branches increases (a phenomenon known as sequence explosion). As the number of sequences increases, the time to complete a simulation run can become prohibitive. One method of overcoming this difficulty is through the performance of conditional runs where a reduced number of branching rules are activated and all other branching rules are suppressed. If the combination of branching rules is selected with care, the analyst should be able to capture a wide range of potential crew behaviors with minimal processing effort. In general, the analyst should select a combination of three or four branching events that reflect high likelihood scenarios. The simulation matrix should ensure that all significant branching rules are explored (i.e., each branching rule should be included in at least one simulation run).

Table 1, Mapping Crew Variability to ADS-IDAC Branching Rules

Type of Variability	Source of Crew-to-Crew Variability	ADS-IDAC Branching Rule(s)
Procedure Execution	Selection of component target control settings (e.g., throttle valve positions, manual controller set points)	Action control value branching rule
	Time required to execute procedure actions	Action time branching rule
	Recovery of failed equipment	Equipment failure and recovery branching rule
	Omission or skipping of procedure steps	Procedure step skipping branching rule. Branching is enabled by the step skipping probability threshold in the operator profile.
	Interpretation of potentially ambiguous procedure criteria (e.g.,, translation of qualitative descriptors such as “decreasing”, “increasing”, or steady” into context-specific quantitative criteria)	No branching rule available. Conditional simulation runs can be run to examine the use of different quantitative thresholds for qualitative criteria.
Operator Mental Models	Training, experience, and diagnostic capabilities	Mental belief branch probability branching rule. The operator knowledge base should include a sufficient spectrum of mental beliefs to capture crew variabilities due to training and knowledge.
	Time required to initiate automatic memorized actions	Mental procedure activation time branching rule
Operator Preferences	Selection of high level goals and problem solving strategies	Troubleshooting probability and procedure use probability branching rules
	Tendency to rely on memorized information rather than control panel readings	Use of memorized information branching rule
	Delays or breaks during procedure execution (e.g., crew briefings)	Action time branching rule. Appropriate procedure hold steps must be incorporated into ADS-IDAC procedures to model delays or breaks.
	Tendency to manually perform anticipatory safety actions prior to automatic actions	Mental belief branch probability branching rule. The operator knowledge base must include appropriate mental belief to trigger the desired actions.

Table 2, Suggested Format for Summarizing Analysis Branching Rules

The following Table summarizes the branching events for a hypothetical complete loss of feed water analysis.

Branch ID	Branching Event	Branching Options	Branching Values	Importance	Comments
TRIP	Action Taker Hardwired Diagnosis: "Complete Loss Of Feedwater"	Branch Probability (2 Branches)	Branch probability options: 0.0, 0.5, or 1.0	High	Enables early reactor trip upon loss of main feed pumps. Branch probability of 1.0 enables early trip, probability of 0.0 blocks early trip, and intermediate value generates both event sequences.
FRGTime	Decision Maker Hardwired Diagnosis: "CSF Loss of Secondary Heat Sink"	Mental Procedure Activation Time Branch (3 Branches)	Time Delay $\alpha = 100.$ $\beta = 1.5$ Long, nominal, and short delay times (3 branches)	High	Sets time delay for initiation of FRG H.1 (once entry conditions have been perceived). Represents crew delay associated with transition from ES 0.1 to FRG H.1
Brief-1	FRG H.1 Briefing Hold 1	Action Time Branches (3 Branches)	Time Delay $\alpha = 200.$ $\beta = 1.75$ Long, nominal, and short delay times (3 branches)	High	Delay time prior to initiation of FRG H.1. Represents crew briefing prior to initiation of FRG H.1 actions
SKIPRCP	Skip FRG H.1 Step 3 (trip reactor coolant pumps)	Skip step threshold value, base step skip probability (2 Branches per pump: perform step and skip step)	Decision Maker "skip action threshold" value: 1.0, 0.75 (skip) Base RCP Skip Probability: $\alpha = 400.0$ $\beta = 10{,}000.0$	Low	Failure to trip reactor coolant pumps will increase net RCS heat input. The RCPSkip base probability and the skip step threshold were increased to avoid the generation of an excessive number of skipped step branches.
STMDUMP	FRG H.1, Step 7.c, excessive or insufficient steam dump rate during secondary depressurization	Procedure Control Value Branches for FRG H.1 Step 7.5 Action "X_Stm_Dump" (3 Branches)	Control Values: 0.25 – Excessive 0.10 – Nominal 0.05 – Insufficient	Medium	Establishes secondary depressurization using the steam dump. Excessive steam dump rate could actuate main steam isolation and delay recovery. Insufficient steam dump rate could delay recovery.

Branch ID	Branching Event	Branching Options	Branching Values	Importance	Comments
MDAFPA MDAFPB	Hardware Reliability: Motor Driven AFW Pump A(B) Activated by actuation of "Reactor_Trip"	Hardware Reliability X_MD_AFW_Pump_A X_MD_AFW_Pump_B (3 branches are generated for each pump: initial success, recovery, and failure)	Trigger Event: Reactor Trip Failure α = 1.0e5 β = 1.0 Recovery α = 1.0 β = 1.0e5	High (established by scenario description)	Failure of motor driven AFW pump A(B) following Rx Trip. Low probability of recovery prevents recovery of pumps.
MFPA MFPB	Initiating Event: Main Feedwater Pump A & B	Initiating Event X_MF_Pump_A X_MF_Pump_B (2 branches are generated for each pump: permanent failure and failure/recovery)	Trigger Event: Reactor Trip Failure α = 1.0e5 β = 1.0 Recovery α = 1.0 β = 1.0e5	High (established by scenario description)	Initiating event is trip of main feedwater pumps. The condensate pump head curve is modified in the RELAP input deck to model degradation. Low probability of recovery prevents recovery of pumps.

Introduction

The current version of ADS-IDAC utilizes the RELAP5/MOD 3.2 computer code to provide a transient simulation of nuclear power plant operation. The RELAP5 code can simulate a wide variety of accident initiators and provides the capability to model key safety systems, controls, and instruments. Advantages of RELAP5 include its proven capabilities as a transient analysis tool and the availability of detailed power plant models. However, due to the intrinsic limitations of the RELAP5 code, it is not currently possible to model core damage states and severe accident scenarios. Consequently, ADS-IDAC is currently limited to the analysis of scenarios up to the start of core damage. Adaption of ADS-IDAC to a more versatile thermal-hydraulic engine, such as the TRACE or MELCOR code, has been identified as a future research activity. However, the inability of RELAP to adequately simulate core damage states is not a significant limitation since the ADS-IDAC crew model is not valid once a nuclear plant activates the emergency response facilities such as the technical support center (TSC) and offsite emergency facility (EOF). Once the TSC and EOF are activate (within approximately one hour of the accident), the dynamics of crew decision making changes dramatically.

The RELAP thermal hydraulic model provides two key functions for ADS-IDAC: (1) the dynamic and realistic representation of plant parameters, and (2) the ability to interact with the thermal hydraulic model by changing component or system operating states. Any parameter value that can be obtained from the RELAP code can be used within the ADS-IDAC environment. The ADS-IDAC control panel links the operator model to the thermal hydraulic model. It should also be noted that the operator cannot instantly use all information available on the control panel. Similar to an actual control room crew, the information must first be perceived by the ADS-IDAC operator model before the information can be used. The ADS-IDAC environment supports a variety of operator control inputs to the thermal hydraulic model including valve position changes, starting or stopping of pumps, safety system actuations, and control system setpoint changes. Four possible control modes can be used for each controllable component: (1) changing the component operating mode (e.g., automatic vs. manual mode), (2) setting a specific control value for a component (e.g., throttling control valve to 50% open), (3) incrementing the control setting of a component (e.g., throttling open a control valve by an additional 10%), and (4) setting a control value based on a perceived parameter (e.g., setting the steam dump target pressure equal to the perceived main steam header pressure). These capabilities provide sufficient flexibility to realistically model all significant operator interactions with the plant model.

The user must provide a description of the nuclear plant model (called an input deck) in order to execute the RELAP code. The input deck is a text file that describes the arrangement of plant equipment; thermal-hydraulic initial conditions and boundary conditions; control and safety systems; and other features of the reactor plant. Although many RELAP plant models have been previously developed to support safety analyses and other regulatory uses, these input decks often require some modification in order to exploit the full capabilities of ADS-IDAC. For example, a RELAP input deck supporting a regulatory safety analysis may conservatively omit certain plant components or features or only include modeling elements for a specific type of accident scenario. In general, an existing RELAP plant model should be modified to include the following features.

- Replacement of all conservative analysis assumptions with realistic best estimate parameters. Typically, conservative values are used for trip setpoints, initial reactor power level, timing delays for automatic safety system actuations, and other key plant parameters. Additionally, significant non-safety control systems and safety features that might not have been included in the RELAP model should be added.
- Modification to safety system models to replace simple thermal hydraulic boundary conditions with a more realistic representation of system components. Often, support and safety systems are simply modeled with a fixed mass flow rate boundary condition in RELAP. These simplified system models should be modified to include redundant subsystems for multi-train systems, provisions for control of significant components such as key pumps and valves, and representation of critical support systems such as water supplies and electrical power.
- Addition of systems and components that provide a significant portion of the mitigative functions provided by the abnormal and emergency operating procedures. The relevant procedures should be reviewed to identify equipment that should be included within the plant model.
- Implementation of control interfaces for all interactive components to allow the ADS-IDAC operator model to manipulate plant systems. This includes the addition of a “manual” control mode for components that normally controlled by an automated system (e.g., feed water regulating valves or power operated relief valves).
- Addition of modeling elements needed to represent initiating event conditions such as coolant or steam leakage paths and equipment failures.

Although a detailed overview of the RELAP thermal hydraulic program is beyond the scope of this manual, the user is encouraged to review U.S. Nuclear Regulatory Commission technical report NUREG/CR- 5535, “RELAP5/MOD3 Code Manual.” In particular, NUREG/CR-5535 Appendix A, “Input Requirements,” provides a comprehensive description of the RELAP input deck.

Detailed Guidance

The following specific guidelines should be used when modifying an existing RELAP plant model:

1. Identify all controllable components and plant indicators that are already included in the original RELAP input deck. Major controllable components generally include valves, pumps, heaters, and the reactor fuel. Plant indicators that are normally present in existing RELAP plant models include: (1) significant inventory/level indicators (e.g., reactor vessel, steam generators, pressurizer, safety injection accumulators), (2) significant mass flow rates (e.g., steam flow, feedwater flow,

reactor coolant flow), (3) significant thermal-hydraulic parameters (reactor coolant system temperature and pressure, steam pressure), and (4) control systems (pressure or inventory control systems). When available, a noding diagram for the plant model can be extremely useful for locating key plant components. Because RELAP input decks are formatted as simple text files, it is also possible to locate many key plant features by searching the RELAP input deck for certain keywords. In particular, the following keywords are often useful for finding major plant components in the RELAP input deck:

TMDPVOL (Time Dependent Volume): Time dependent volumes are used to model a mass source or sink using a controlled pressure, temperature, and/or steam quality condition. The mass flow rate into the TMDPVOL is adjusted accordingly. This hydraulic component is often used to model exhaust volumes for relief valves or leakage paths or inexhaustible sources of water.

TMDPJUN (Time Dependent Junction): Time dependent junctions are used to model a mass source or sink using a controlled mass flow rate boundary condition. The thermal-hydraulic conditions across the junction are adjusted accordingly. These hydraulic components are often associated with safety injection system connections, makeup or letdown systems, or inventory control systems.

ANNULUS (Annulus Hydraulic Component): The annulus hydraulic component is often used to model the reactor vessel or steam generator down comer volume. Water levels for these components are often measured in the annulus region.

PRIZER (Pressurizer): The pressurizer component is used to model the pressurizer in a pressurized water reactor. Significant control components such as the spray valves and relief valves are usually connected to the pressurizer hydraulic volume. The pressurizer pressure and level are also significant control panel indicators.

VALVE (Valve): As the name suggests, the valve hydraulic component is used to model flow limiting devices. Valve components are often used to control thermal hydraulic parameters or initiate accident conditions. RELAP allows the user to select from a variety of valve types, including check valves, relief valves, and remotely operated valves. The following key words are often useful in locating specific valve types:

- TRPVLV (Trip valve): Trip valves have only two operating states: fully open or fully closed. Trip valves fully open when the associated logical or variable trip (specified in card CCC0301) is on, the valve is fully opened. The valve repositions immediately upon a change in state of the trip variable. This component is often used to model system leaks or ruptures.

- MTRVLV (Motor valve): Similar to the trip valve, motor valves have two normal operating states – fully open and fully closed. However, two trip variables are specified in card CCC0301, an open trip and a closed trip. A rate of valve position change is also specified in card CCC0301 to define that the time it takes for the valve to move from the fully open to fully closed position. As such, the motor valve does not immediately reposition, but can be in an intermediate state for a finite length of time. This component type is typically used to model large isolation valves such as main steam stop valves, turbine trip valves, and feed water isolation valves.

- SRVVLV (Servo valve): a servo valve can be positioned fully closed, fully open, and any intermediate position. The valve position is determined by the normalized flow area set by the control variable entered into card CCC0301. Servo valves are typically used to model throttle valves and are often used as power operated relief valves or to control mass flow rates. The pressurizer power operated relief valve, the steam generator atmospheric steam dump valves, and the feed water regulating valves are usually modeled with a servo valve hydraulic component.

PUMP (Pump): As the name suggests, the pump hydraulic component is used to model pump components. Typically, this component type is only used for large pumps where coast down flow following a pump trip is significant. Examples include reactor coolant pumps and feed water pumps. Smaller pumps are usually modeled with time dependent hydraulic junctions that establish a mass flow rate based on the differential pressure across the junction. Pump components may also include reference to a logical or variable trip that can initiate a pump countdown (card CCC0301).

ACCUM (Accumulator): The accumulator hydraulic component is used to model a liquid/gas accumulator. Typical applications include the passive safety injection accumulators in a pressurized water reactor.

2. Identify all automatic control systems included in the original RELAP input deck. The RELAP plant model will usually include several automatic control systems to maintain key thermal hydraulic parameters at a predefined setpoint. Typical examples include pressurizer level control, reactor coolant pressure control, feed water control, and steam pressure control systems. In addition to improving the realism of the plant model, control systems enhance the ability of the thermal hydraulic model to reach a stable steady state condition. In the absence of a control system, small errors in initializing the thermal hydraulic model (such as a mismatch between steam and feed water flow rates) can cause significant plant deviations over time. Control systems mitigate these initialization errors and allow the model to achieve a stable equilibrium condition. Control systems can be found in the 20500000 card series in the input deck and are usually associated with either the **FEEDCTL** (feed water control) or **PROP-INT** (proportional integral control)

controller type. The feed water control component is usually used to control reactor vessel or steam generator water level. Most other control systems use a proportional-integral controller to increase the capability to maintain a target parameter on the established set point. Because control systems must eventually actuate a controllable component, control systems can also be identified by back-tracing the control variable for the associated component (e.g., the control variable associated with the pressurizer pressure control logic will be referenced in the servo valve description for the spray valve). At a minimum, automatic control system should be provided for the following pressurized water reactor control functions:

- Pressurizer pressure control (spray valves, power operated relief valve, and heaters)
- Pressurizer level control (reactor coolant system makeup and letdown flow)
- Steam generator level control (feedwater regulating valve)
- Steam generator pressure control (atmospheric relief valves)
- Steam header pressure control (condenser steam dump valves)

Boiling water reactor models will have a similar set of control functions and actuating devices. Once automatic control systems are located, several other key RELAP model features can usually be identified. These include key thermal hydraulic parameters (such as reactor coolant system pressure or key water levels) and controller set points.

3. Modify automatically controlled components to permit to permit the ability to manipulate equipment from the ADS-IDAC environment. In general, nuclear plant automatic control systems allow the operators to take manual control of the final actuating device(s) or change the control set point. In addition to allowing more realistic plant control, the manual control mode can also be used to simulate hardware failures and initiating events. Ideally, three modes of operation should be provided for each automatic control system:

   i. fully automatic - the control system positions the final actuating device(s) to maintain target parameter at the nominal setpoint,
   ii. fully manual control - the operator positions the final actuating device(s), and
   iii. setpoint control - the control system positions the final actuating device(s) to maintain the target parameter at the user selected setpoint.

   To illustrate the general methodology for adapting an automatic control system in an existing RELAP input deck for use with ADS-IDAC, consider a feedwater control system for a pressurizer water reactor. Based on the relative mismatch between main steam and feedwater flow rates and water level error (i.e., the difference between the actual steam generator water level and the level setpoint), the control system generates a normalized valve position for the feedwater regulating valve (FWRV). The FWRV controls the mass flow rate into the steam generator and is usually modeled using a servo valve hydraulic component or a time dependent junction. The following procedure should be used to adapt the existing control system for ADS-IDAC:

a) Set up an interactive control variable in the RELAP input deck to permit ADS-IDAC to pass FWRV control settings to the thermal-hydraulic model. Interactive control variables are specified in cards 800 – 999 of the RELAP input deck. The following input deck card sets up the interactive variable "sgcfrv" in input deck card 810 with an initial value of -1.0:

```
0000810   sgcfrv   -1.0   $ SG C feedwater reg valve
```

b) Add a variable trip that will enable the control system to be toggled between automatic and manual mode. One method of accomplishing this function is to enable the automatic control mode when the associated interactive variable is negative and use positive control variable values as the manual control setting. The following card sets up variable trip 475 as a toggle control between automatic and manual control modes (when the trip is true, the control system will be in automatic mode):

```
20604750  sgcfrv  1000000000 gt  null   0    -0.01   n
```

In this case, a threshold value of -0.01 is used to allow the use of 0.0 as a manual control input for the fully closed position. The "n" specifies that the variable trip does not latch and can be reset as the simulation progresses.

c) Adjust existing control system to add manual control capability. Assuming that control variable card 725 was previously used as the output of the control system, the following additional cards allow either a manual or automatic control signal to be sent to the final actuating device:

```
20572600 "sgcfwman" tripunit  1.0  0.0  1  3  0.0 1.0
20572601   475
20572700 "sgcfwaut"   sum     1.0  1.0  1  3  0.0 1.0
20572701   1.0       -1.0     cntrlvar 726
20572800  "sgcfw_mc"   mult    1.0  1.0  1  3  0.0 1.0
20572801  cntrlvar  726        sgcfrv    1000000000
20572900  "sgcfw_ac"   mult    1.0  1.0  1  3  0.0 1.0
20572901  cntrlvar  727        cntrlvar 725
20573000  "sgcfw_tc"   sum     1.0  1.0  1  3  0.0 1.0
20573001  0.0   1.0  cntrlvar 728  1.0   cntrlvar 729
```

The basic approach is to use a tripunit control variable to toggle between automatic and manual mode (cards 20572600 and 20572700). Since the output from the tripunit function can be only 0.0 if the trip is false or 1.0 if the trip is true, variable control cards 20572600 and 20572700 are complimentary in that one is always equal to 1.0 and the other is equal to 0.0. The manual control signal is calculated in control variable card 20572800 and is equal to 0.0 if the control mode is in automatic or the value interactive variable sgcfrv if the control mode is manual. Similarly, control variable 20572900 is the automatic control signal and is equal to either 0.0 (if the control mode is in manual) or the automatic control signal from control variable 725 (if the control mode is in automatic). Since only control variable 20572800 or 20572900 can be non zero (i.e., at least of the control variables is equal to 0.0), summing these two values together selects the appropriate control signal. The final value of control variable 730 is used to control the FWRV throttle position and is limited to the range 0.0 – 1.0 (0.0 is fully closed, 1.0 is fully open).

d) Revise control variable input for final actuating device(s) to use the new control input. Assuming that the FWRV was hydraulic component 500, card 5000301 would be used to assign new control variable 730 for positioning control:

```
5000301        730
```

In a RELAP input deck, it is possible to enter multiple versions of the same card. If more than one version of the same card number is present, RELAP uses the last card version listed in the input deck. Therefore, the additional RELAP input deck coding can be placed at the end of the RELAP input deck to preserve the integrity of the original plant model (i.e., it is not necessary to delete or modify coding in the original RELAP deck since all revisions needed to support ADS-IDAC can be placed at the end of the input file).

4. Add interactive controls. Based on a review of available plant procedures, operator behavior rules, training materials, and other similar data sources, the analyst should determine what plant control functions should be included in the RELAP thermal-hydraulic model. In general, any plant component that provides a substantial mitigative function following an abnormal event or an accident should be controllable by ADS-IDAC. ADS-IDAC controls these RELAP components using the interactive control variables specified in cards 800 – 999. Three possible control modes are available in ADS-IDAC: (1) setpoint adjustment for automatic control systems (e.g., set steam dump setpoint to the perceived steam header pressure), (2) changing component states (e.g., run/stop), and (3) fine adjustments to a component control position (e.g., changing throttle valve position). Fine adjustments can be based on an absolute control demand (e.g., position a control valve to the 50% open position) or a relative demand (e.g., open a control valve an additional 10% from the current position). In order to utilize any of these control modes, appropriate interactive control variables must be specified in the RELAP input deck and cross referenced in the ***RELAP_channels.txt*** input file (see Section 3). When specifying interactive variables, it is important to note that the second word of each card is the default value of the variable. The default value should be set to the normal full power value for the plant (nominal setpoints, control systems in automatic, standby systems aligned for automatic start, etc.). Examples of possible interactive control variables include:

Setpoint Control

Setpoint control allows ADS-IDAC to change the controlling setpoint used in a RELAP automatic control system. In this example, interactive variable 801 enables control of the steam generator atmospheric relief valve (with a default setpoint of 1040 psia) and interactive variable 802 enables control of the condenser steam dump system setpoint (with a default setpoint of 1020 psia):

0000801	***sgcporvp***	***1040.0***
0000802	***stmdumpp***	***1020.0***

The control system description for each of these control systems (card series 205xxxx00) would also need to be modified to use the new interactive variables to determine the control signal for the final actuating device.

Component Operating State

Operating state control (called "Control_Panel_Controller" in the ***ControlPanel.txt*** input file), provides simple state control for components. These controls generally apply to binary state components (e.g., on/off, open/closed). In this example, interactive variable 803 enables control of the reactor trip function while variable 804 enables control of the turbine trip function:

0000803	***scram***	***-1.0***
0000804	***turbtrip***	***-1.0***

Assuming the control system is designed such that the trips are activated when the variable value is greater than 0.0, these variables are initialized to a value less than zero to place the trip functions in standby.

Fine Adjustment Control

Fine adjustment control (called "Control_Panel_Fine_Adjust" in the ***ControlPanel.txt*** input file) provides a continuous range of possible control settings for the associated component. This type of interactive variable is usually used to throttle or regulating valves. In this example, interactive variable 805 is used to control the pressurizer spray valve while variable 806 is used to control the pressurizer power operated relief valve (PORV):

0000805	***spray***	***-1.0***
0000806	***pzrporv***	***-1.0***

Both the spray valve and the PORV serve as the final actuating device of a pressure control system and can be operated in either automatic or manual mode. The initial value of -1.0 initially places the control system in automatic operation. When the variable value is equal to or greater than 0.0, the control system is placed in manual and the valve is positioned to a stem position equivalent to the variable (e.g., a variable value of 0.5 positions the valve in manual mode to 50% open).

Once appropriate interactive control variables are specified, the analyst must link these variables into the thermal-hydraulic model by appropriate references in control variable, hydraulic volume, or heat structure descriptions..

5. Improve overall plant level control for normal operation and abnormal conditions. The RELAP thermal hydraulic program is generally used to perform accident analyses for nuclear power plants. Consequently, components, equipment, and systems normally used during power operation are often omitted or simplistically modeled in the input deck. For example, if a turbine model is included, the turbine load could be set at a full power value with no means available to adjust turbine load. Similarly, existing RELAP models for non-safety control systems such as makeup and letdown functions may not provide a sufficient range of flexibility to model an adequate range of operator interactions. Therefore, the analyst should review the RELAP input deck and identify any features that need to be added to provide sufficient capability to model operator behaviors that might occur during power operation. Typical examples include:

    - turbine generator load control
    - non-safety related interlocks or protective features (e.g., turbine runback)
    - charging/letdown system operation
    - nuclear reactivity control systems (e.g., control rods, emergency boration)
    - condenser steam dump control

6. Activate the nuclear reactor core point kinetics model. RELAP provides the two main methods to control the reactor power level: (1) specifying a time dependent power level using either a control variable or tabular input vale, or (2) calculating reactor power level based on a point kinetics model. When the first method is used, a constant (and conservative) power level is used prior to actuation of a reactor trip. Following a reactor trip, reactor power follows a time dependent decay curve. Although this method provides relatively straightforward means to control reactor power level, the effects of important feedback mechanisms such as reactor coolant temperature changes and control rod motion cannot be easily modeled. Activation of the point kinetics RELAP option allows better modeling of reactivity feedback mechanisms and provides a more realistic model of the plant response following an accident or control input. In order to use the point kinetics model, it is necessary to locate the nuclear fuel heat structures in the input deck. Heat structure components are used to model heat sources and sinks and can be found in the 1CCCGXNN card series[7]. The heat source data cards (1CCCG701 – 1CCCG799) specify how the power output of the nuclear core will be determined. The heat output can be

[7] The numbering scheme for heat structure components is interpreted as follows: CCC refers to the heat structure number, G refers to the geometry number and is used to identify different types of heat structures (such as fuel pins and core barrel), X is the card type and NN is the card number within a card type.

determine from a general table (card series 202TTT00), a control variable (card series 202CCC00), or from the point kinetics model.

If the point kinetics model will be used, the analyst is required to provide core kinetics data in the 30000000 card series of the input deck. Parameters such the temperature and density reactivity coefficients, core power shaping factors, scram reactivity worth, and delayed neutron lifetimes can be adjusted within the RELAP model. In order to improve model realism and stability, it is also recommended that the following system functions be added to the RELAP model:

- Control rod reactivity
- Emergency boron addition
- Safety injection boron addition

Each of these systems can be added by creating a control system (205CCC000 card series) to represent the amount of reactivity added by system actuation[8]. The control system can then be linked to the point kinetics model in cards 30000011 – 30000020 (control variable reactivity feedback). It is also recommended that a temperature control system be developed to automatically add or subtract control rod reactivity to maintain the reactor coolant system at the programmed value.

The analyst should be aware that activation of the point kinetics model can result in unexpected stability issues. If problems are encountered, the analyst should reduce the maximum allowable RELAP time step (cards 201-299) or ensure that sufficient negative reactivity is added following a plant scram or safety injection actuation. Increasing the scram rod worth or post safety injection boration will reduce the potential for a core restart event (and the associated model instability) due to reactor coolant system cooldown.

7. Enhance existing system models to provide realistic controls, alarms, and indicators. In general, front line safety systems are modeled in existing RELAP input decks using only a time dependent junction at each fluid injection point. For example, the high pressure safety injection system for a three loop pressurized water reactor is usually modeled using only three time dependent junctions to provide injection flow to each reactor coolant loop (Figure 10). Although this modeling is often sufficient for the purposes of a deterministic safety analysis, the simple model lacks the major control elements, alarm functions, and indications found in the actual plant system. In order to improve the realism of the model, it is necessary to modify the system models to add multiple trains, control valves, and piping elements.

[8] Although RELAP is capable of modeling boron concentration, it is not recommended that this option be used within the ADS-IDAC environment. The effects of boron concentration changes and control rod motion can readily be modeled with control variables.

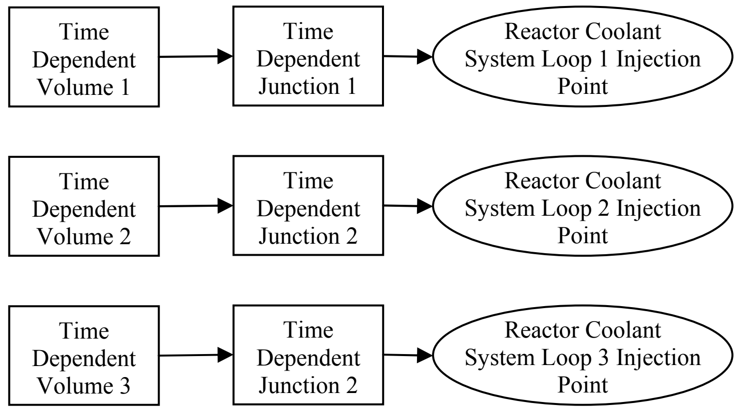


Figure 10, "Typical RELAP Mitigating System Model"

The modified system model includes separate subsystem trains for each injection pump and realistically models potentially hydraulic dependencies for loop injection flow (Figure 11). The benefits of the modified system model include:

- Capability to independently control of multiple injection pump trains (time dependent junctions A and B) and individual loop flows (loop servo control valves).
- More realistic representation of actual pump head/flow characteristics since the improved model more closely matches the actual plant configuration
- Improved modeling of injection flow dependencies between the reactor coolant loops. Since the injection flow is supplied from a common header (the common hydraulic volume), the improved model does not decouple the loop injection to one loop from the other loops (e.g., high injection flow to loop 1 will decrease injection flow to loops 2 and 3).

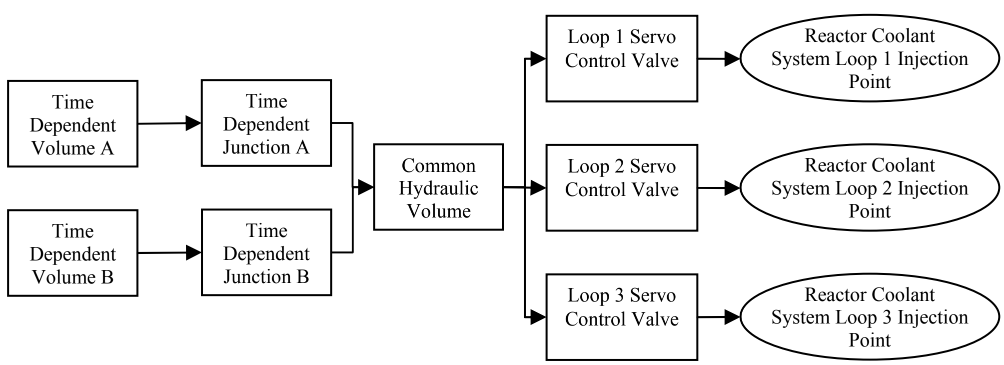


Figure 11, "Modified ADS-IDAC Mitigating System Model"

It is recommended that the analyst review the system level modeling for all front line safety systems in the existing RELAP input deck (e.g., high pressure safety injection, low pressure safety injection, and auxiliary feedwater). If an overly simplified modeling approach is used, the safety system models should be revised to more closely approximate the actual configuration of the system.

8. Add support system dependencies. In general, RELAP plant models focus do not include detailed models for support systems such as electrical power, cooling water, or lubricating oil. These support system dependencies can be modeled by adding appropriate trip variables for key components. For example, a loss of offsite electrical power can be modeled by setting up an interactive and trip variable that can be used to toggle the availability state of offsite power. The trip variable for loss of power can then be referenced in the trip logic for individual components that would normally be lost following a loss of power. Typical components powered from offsite power include reactor coolant pumps, turbine-generator auxiliaries, and condenser circulating water. A similar procedure can be used for other support systems such as cooling water or lubricating oil. In this manner, support system dependencies can be simulated without the need for developing a detailed system model.

9. Create derived or unique control panel indicators. ADS-IDAC includes capabilities to build control panel indicators for basic thermal hydraulic parameters such as pressure, temperature, and flow. However indicators that are derived from basic parameters (e.g., average reactor coolant temperature or subcooling margin) or that rely on unique parameters must be added directly into the RELAP input deck. A sampling of the unique parameters can be obtained from the RELAP plant model include:

   - ***ACVLIQ*** - liquid volume in the referenced accumulator tank, standpipe, and surge line.
   - ***VLVSTEM*** - ratio of the current valve stem position to the fully open valve stem position for the referenced motor and servo valves
   - ***HTCHF*** – critical heat flux for the referenced heat structure.
   - ***HTCHFR*** - critical hat flux ratio for the reference heat structure.
   - ***HTMODE*** - heat transfer mode number (i.e., heat transfer regime in effect) for the referenced heat structure.
   - ***RKRECPER*** - reciprocal reactor period (inverse seconds) calculated from the point kinetics model.
   - ***RKTPOW*** - total reactor power calculated from the point kinetics model.

   These unique control panel indicators are used by setting up a control variable capable of reading the parameter of interest. For example, if hydraulic component 350 is a servo valve, the following control variable can be used to indicate the valve position:

```
*ctlvar      name       type    factor   init   f c   min    max
20541500   "vlv_pos"    mult      1.0     0.0   0 3   0.0    1.0
*ctlvar    variable name  parameter no.
20541501      vlvstem          350
```

This control variable could be added to the ADS-IDAC control panel by referencing control variable 415 in the ***ControlPanel.txt*** input file (see Section 3 for additional information on this input file).

A similar process is used to calculate derived parameters for the ADS-IDAC control panel. The analyst should aware that parameters obtained directly form the RELAP model will be in SI units. Therefore, it will be necessary to adjust the control variable output if British parameter units will be used. For example, the following control variables can be used to calculate the average fluid temperature of hydraulic volumes 150, 250, and 350 in degrees Kelvin and Fahrenheit:

```
*ctlvar      name       type    factor   init   f c   min    max
20529000  "avgTempK"     sum    0.333    550.0  1 0
*ctlvar    constant  scale   variable name  parameter no.
20529001     0.0      1.0       tempf         150010000
20529002              1.0       tempf         250010000
20529003              1.0       tempf         350010000

*ctlvar      name       type    factor   init   f c   min    max
20529100  "avgTempF"     sum     1.0     530.0  1 0
*ctlvar    constant  scale   variable name  parameter no.
20529101   -459.7     1.8      cntrlvar         290
```

Control variable 290 calculates the average of the fluid temperatures in the three hydraulic volumes. Because the tempf function returns the fluid temperature in SI units, the output for control variable will be in degrees K. Control variable 291 is used to convert the output from control variable 290 to degrees F.

10. Expanded trips. Trips are Boolean variables provide alarm functions, activate safety systems, and support control system operation. RELAP provides two types of trip devices – variable trips and logical trips. Variable trips change state when a target parameter exceeds a preset threshold. Logical trips are used to combine two or more variable and logical trips to build a logical expression using Boolean operators. Although RELAP provides fixed number of variable and logical trips, the maximum number of available trip variables can be increased when an expanded trip option is selected. When the default option is used (i.e., the expanded option is not used), variable trips must be specified in cards 401-599 and logical trips in cards 601-799. Therefore, the default option provides a maximum of 199 variable trips and 199 logical trips. Unfortunately, this number of trip variable might be insufficient to adequately model the reactor plant for use with ADS-IDAC. If this occurs, the analyst should consider the use of the expanded trip option by entering the word “expanded” on input card 20600000. The expanded option changes the variable trip card range to 20600010-20610000 and the logical trip range to 20610010-20620000. The expanded option increases the number of available variable and logical trips up to a maximum of 1000, each. Transitioning the input deck from the default to expanded trip options will require that all existing trips be renumbered in expanded trip format:

- Add the following card to the RELAP input deck to activate the expanded option:

20600000 expanded

- If the developers of the original RELAP deck entered trip variable card numbers using an eight digit format (e.g., trip 450 is entered in card 00000450), the transition to an expanded format only requires that replacement of the first four digits of the card number with the three digit prefix "206" and the addition of a trailing zero. For example:

00000<u>450</u> ⟶ **2060<u>450</u>0**

Default Card Number *Expanded Card Number*

In this case, the designation of the variable trip remains the same under when expanded numbering is used. When transitioning to expanded format, references to the variable trip number do not need to be revised elsewhere in the input deck (i.e., the default variable trip 450 is still trip 450 when the expanded option is chosen). All variable trip cards (card numbers 401-599) must be revised to the expanded card numbering format or a RELAP error will occur.

- Modification of logical trips is more complicated because the trip number needs to be revised when the expanded option is selected. Specifically, all expanded logical trips numbers must fall in the range 1001 – 2000. Because the default option requires logical trips to be numbered from 601 – 799, the default trip numbers will no longer be valid logical trips under the expanded option. The easiest way to transition logical trips is to add 1000 to the default logical trip number (i.e., logical trip 650 will become trip 1650 in the expanded format):

00000<u>650</u> ⟶ **206<u>1650</u>0**

Default Card Number *Expanded Card Number*

Unfortunately, it will also be necessary to change every reference to the default logical trip throughout the RELAP input deck to reflect the new trip number. Although a simple word search can be conducted to facilitate these modifications, the analyst should be aware of the following:

  - A three digit integer in the RELAP input deck can refer to either a control variable or a trip variable. Only logical trip variables should be changed or unexpected errors will occur. Therefore, the analyst should carefully review the context that trip number is used to ensure that the correct variable type is being changed.

- ➢ Trips can be referenced either by its integer trip number or by its complement. Thus, if default logical trip 650 was used in the original input deck, the analyst should change all references to trip "650" to "1650" and all references to its complementary trip "-650" to "-1650".

The expanded option should provide a sufficient number of variable and logical trips to adequately represent even complex reactor plant models.

11. Time step control. The RELAP end time limit and time step is controlled by input cards 201-299. The RELAP end time limit should be set to a value greater than the ADS-IDAC sequence truncation time to prevent RELAP from prematurely terminating the simulation. The maximum time step should be set to the highest value that permits stable model behavior. If the RELAP model runs slowly, the maximum time step can be increased or a different time step control option can be selected (e.g., implicit or nearly implicit time step control).

Section 3: ADS-IDAC File Structure

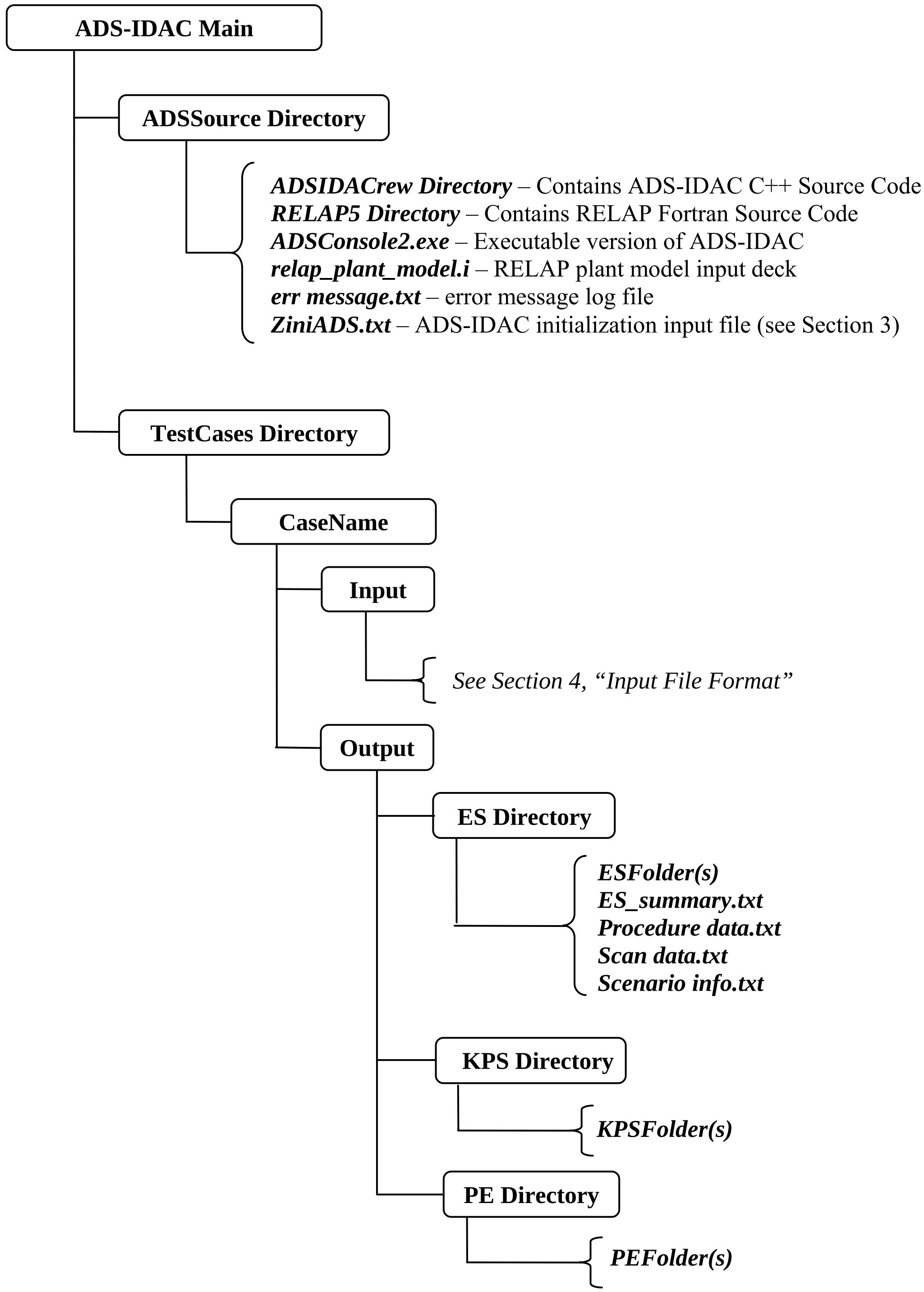


Figure 12, "ADS_IDAC Directory and File Structure"

ADS-IDAC File Structure

ADS-IDAC is supported by a relatively complex input file structure. The various input files provide data associated with the nuclear plant model, operator knowledge bases, and program executive control (e.g., branching and sequence truncation rules) needed for code execution. ADS-IDAC input files are described in detail in Section 4. The results of an ADS-IDAC simulation are summarized in several output files. These files provide data about nuclear plant thermal-hydraulic behavior, operator cognitive decision-making, and information required to reconstruct the dynamic event tree structure. A narrative summary of each sequence is also provided. The remainder of this section describes the ADS-IDAC output files.

Output Files

Three general categories of output files are generated by ADS-IDAC: event sequence descriptions, key parameter data files, and branch information. The output files are all written in plain text file format and can be imported into a third party program such as MS Excel for data analysis and visualization. The output files are located in the following file folders within the output file directory (see Figure 12):

ES File Folder

- ESFolder***_folder_number***: The end state (ES) output is arranged into individual file folders. ADS-IDAC limits the number of individual files within an output folder to less than 1000 files. Therefore, if more than 1000 output files are generated by a simulation, additional ESFolders are created to store the data. The first 1000 output files are stored in ESFolder_0, the seconds 1000 files are stored in ESFolder_1, and so on. For each end state generated during the simulation, a unique ZESNode_***endstate_number***.txt file is placed in the appropriate ESFolder to provide a detailed description of the sequence end state.

- ES_summary.txt – Summarizes all sequences associated with the simulation. For each sequence, the sequence length, probability, and termination criteria are specified.

- Procedure step.txt – Summarizes data associated with the ADS-IDAC procedure step skipping module. Output data includes:
    i. Time
    ii. Operator
    iii. Procedure and step number
    iv. Action
    v. Relevance of action to operator's situational assessment
    vi. Time Constraint Loading for the associated operator
    vii. Dynamic factors associated with step skipping model
    viii. Static factors associated with step skipping model
    ix. Probability of skipping associated step action (error of omission)

The ADS-IDAC step skipping model is described in Section 4, "Procedures.txt".

- Scan data.txt – Summarizes the data associated with the ADS-IDAC control panel scanning module. Output data includes:
    i. Time
    ii. Operator
    iii. Size of the associated operator's scan queue (i.e., the total number of control panel items scanned)
    iv. Contents of the operator's scan queue (i.e., specific alarms, components, and parameters included in the control panel scan)

  The control panel scanning model is described in Section 4, "KB_AOT_Scanned_Parameters.txt".

- Scenario info.txt – Provides a detailed description of all sequences, including all branching points and associated events.

<u>KPS File Folder</u>

KPSFolder***_folder_number***: The key parameter state (KPS) output is arranged into individual file folders. ADS-IDAC limits the number of individual files within an output folder to less than 1000 files. Therefore, if more than 1000 output files are generated by a simulation, additional KPSFolders are created to store the data. The first 1000 output files are stored in KPSFolder_0, the seconds 1000 files are stored in KPSFolder_1, and so on. For each sequence, the following output files are created:

- ZDIA_***sequence_number***.txt – For each operator, provides the output from the diagnostic engine. Output consists of a time history of the diagnosis event confidence level for each operator.
- ZKPS_***sequence_number***.txt – Provides a time history of all indicators specified in the ***ControlPanel.txt*** input file.
- ZPIF_***sequence_number***.txt – For each operator, provides a time history of the dynamic PIF values for system criticality, time constrained loading, and information loading.

<u>PE File Folder</u>

PEFolder***_folder_number***: The pivotal event (PE) output is arranged into individual file folders. A sequence consists of a series of connected PE nodes. Similar to the KPS file folder, ADS-IDAC limits the number of individual files within an output folder to less than 1000 files. Therefore, if more than 1000 output files are generated by a simulation, additional PEFolders are created to store the data. The first 1000 output files are stored in PEFolder_0, the seconds 1000 files are stored in PEFolder_1, and so on. The collection of PENode data files provides sufficient information to reconstruct the DDET for the simulation. For each PE node, the following output file is created:

- ZPENODE_***node_number***.txt – Specifies the type of branch event, the time of the event, and the sequence number associated with the PE node.

(this page intentionally blank)

Section 4: Input File Format

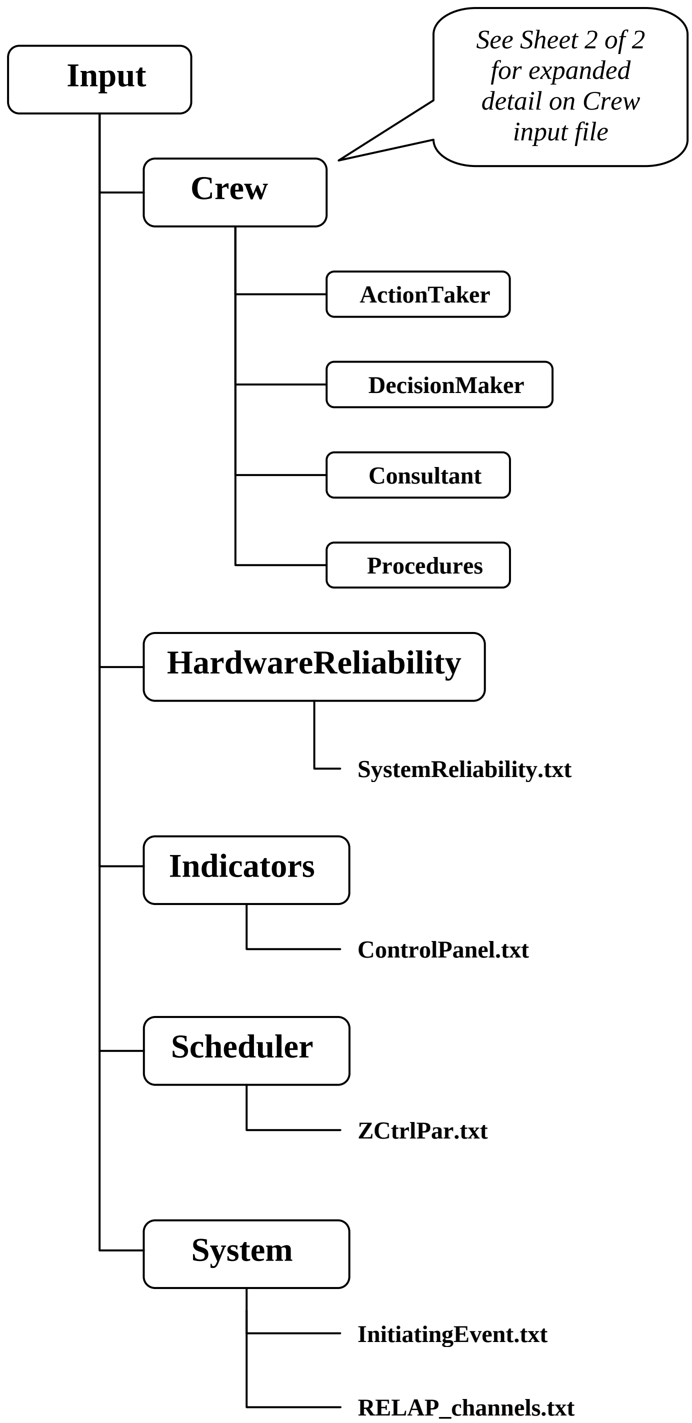


Figure 13, "ADS-IDAC Input File Structure (Sheet 1 of 2)"

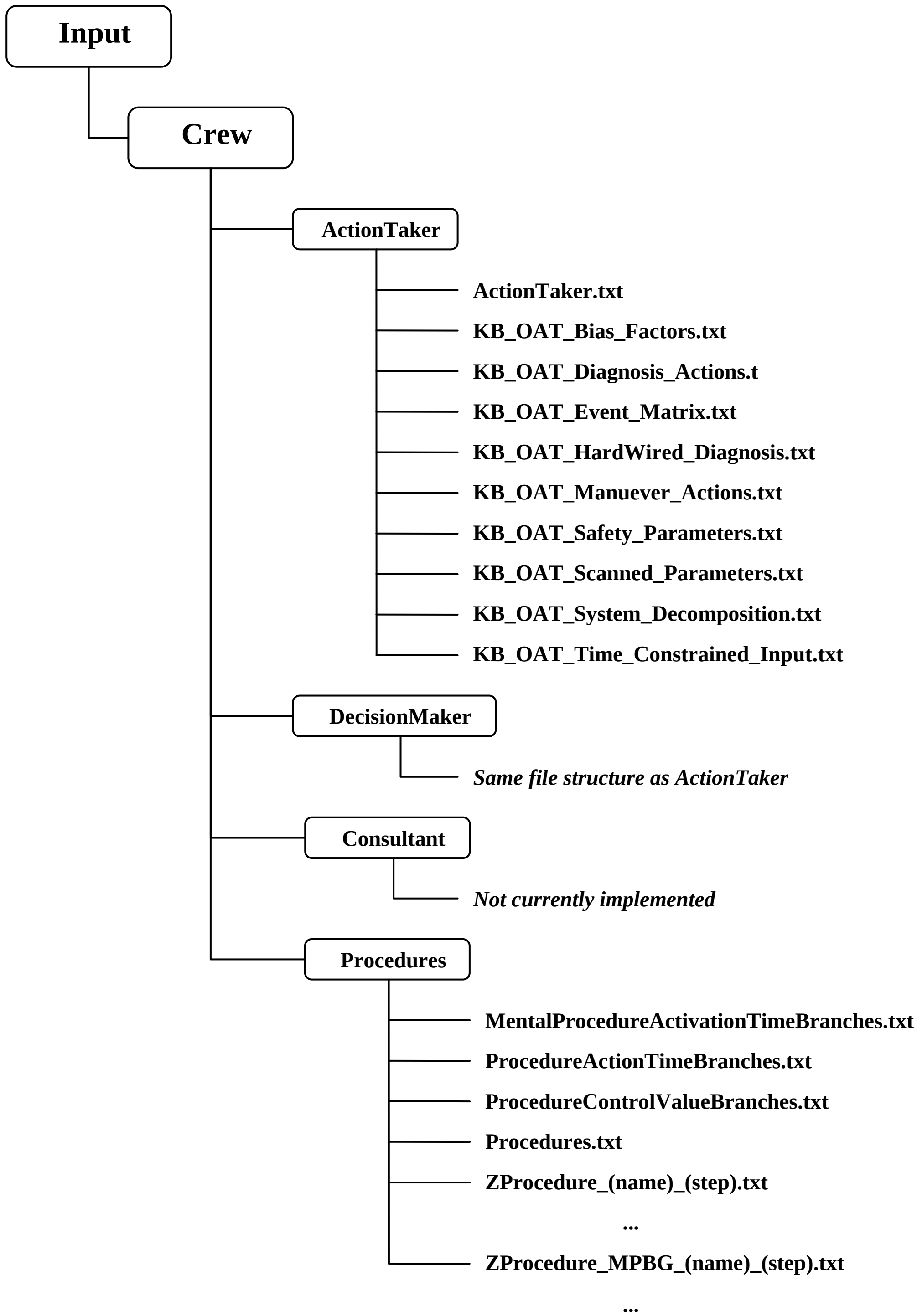


Figure 14, "ADS-IDAC Input File Structure (Sheet 2 of 2)"

Operator Name: Action Taker (OAT)

Responsibilities:

The Action Taker fulfills the role of reactor operator within the nuclear plant control room environment. All crew interactions with the reactor plant model are executed by the Action Taker through use of the appropriate control panel indicators, alarms, and controls. The Action Taker generally follows the direction of the Decision Maker, but may independently execute certain skill-based memorized actions. Typical self-directed Action Taker activities include control of auxiliary feed water following a reactor trip, initiation of reactor trip or safety injection signals during degraded plant conditions, and control panel monitoring in response to plant alarms. Although the Action Taker decision-making processes include goal and strategy selection, these processes generally follow the direction of the Decision Maker (e.g., the Action Taker will set their high level goal equal to the Decision Maker's goal). These input files, in conjunction with the procedure step input files, constitute the knowledge base for the Action Taker. The knowledge base is intended to mode the skills, abilities, and experience of the operator.

Input Files:

ActionTaker.txt
KB_OAT_Bias_Factors.txt
KB_OAT_Diagnosis_Actions.txt
KB_OAT_Event_Matrix.txt
KB_OAT_HardWired_Diagnosis.txt
KB_OAT_Maneuver_Actions.txt
KB_OAT_Safety_Parameters.txt
KB_OAT_Scanned_Parameters.txt
KB_OAT_System_Decomposition
KB_OAT_Time_Constrained_Input.txt

1. Purpose

The "ActionTaker.txt" input file is used to specify certain static behavior factors that shape operator behavior. These factors are generally associated with operator tendencies to pursue certain goals and strategies, the use of perceived information, and the time required to perform activities. This file also specifies several factors that influence the generation of branching events for step skipping and goal selection.

2. Input File Format

```
Action_Taker                          number
action_time_multiplier                time_multiplier
confidence_level_for_acting_HWKB      minimum_confidence
use_memorized_info
information_branch_probability
initial_scan_queue_limit              scan_queue_limit
lower_info_load_threshold             info_lower_threshold
upper_info_load_threshold             info_upper_threshold
alarm_update_time                     double_1    double_2
double_3
component_update_time                 double_1    double_2
double_3
parameter_update_time                 double_1    double_2
double_3
skip_action_threshold                 skip_action_threshold
skip_non_response_threshold           skip_non_response_threshold
abnormal_signal_threshold             abnormal_threshold
mental_proc_priority_threshold        priority_threshold
nominal_communication_time            communication_time
troubleshooting_probability
troubleshooting_branch_probability
procedure_use_probability
procedure_use_branch_probability
```

3. Input Description

number: Format: Integer. Formally used to identify number of operator behavior factors included in input file. This parameter is not used in current version of ADS-IDAC - enter a dummy integer value (e.g., "1").

time_multiplier: Format: Double. Range: > 0.0. This parameter used to proportionally adjust the operator execution time for communication, action, and decision-making activities. The action time multiplier is uniformly applied to all operator activities. A factor of 2.0 doubles the activity execution time compared to the baseline time while a factor of 0.5 reduces the activity execution time by a factor of ½. No dynamic event tree branches are generated by this parameter.

minimum_confidence: Format: Double. Range: 0.0 – 1.0. This parameter is not used in the current version of ADS-IDAC – enter a dummy double value (e.g., 0.0). The confidence level for activating operator hard wired diagnoses is specified in the HardwiredDiagnosis.txt input file within the operator knowledge base.

information_branch_probability: Format: Double. Range: 0.0 – 1.0. This parameter establishes the branching probability for enabling the operator's use of previously perceived (and memorized) plant data. When the use of memorized information is enabled, the operator will use of memorized information (if available and current) to address data requirements of procedure expectations and knowledge-based action prerequisites. When the use of memorized information is blocked, the operator will always obtain current information from the plant control panel. The use of memorized information can reduce activity execution time but may result in the use of outdated and incorrect information. For values greater than 0.999999, the use of memorized information is always enables. For values less than 0.000001, the use of memorized information will always be blocked (i.e., the operator will always obtain recent information form the control panel when evaluating procedural expectations or knowledge-based action prerequisites). Intermediate values will cause a branching point to be generated early in the simulation where one branch enables the use of memorized information (with the branching probability set to the input value) and a second branch blocks the use of memorized information (with the branching probability set to the complement of the input value). Even when the use of memorized information is enabled, the operator may block the use of previously perceived information if it is not recent. The criteria used to judge the recency of plant data is established in the alarm, component, and parameter update time input parameters.

scan_queue_limit: Format: Integer. Range > 0. This parameter sets the limit of the maximum number of parameters that may be placed in the operator's scan queue. The operator periodically updates the memorized values of parameters contained in the scan queue with recent information from the control panel. A higher scan queue limit will allow the operator to monitor more parameters and obtain an improved situational assessment of the plant state. Setting a lower scan queue limit reduces the number of parameters that can be periodically monitored and allows the analyst to simulate the operator's information processing and short term memory limitations. The actual scan queue limit is dynamically adjusted during the simulation and may be less than this input value due to the influence of certain performance influencing factors. When the size of the scan queue exceeds the dynamic limit, low priority parameters are removed form the queue until the size limitation is met.

info_lower_threshold: Format: Integer. Range: > 0. The lower and upper information load thresholds are used to calculate the value of the information load performance influencing factor. The information load PIF is based on the operator's average information processing rate. When the information processing rate is less than the lower threshold, the PIF value is set to 0.0. When the average information processing rate is greater than the upper threshold, the PIF value is set to 10.0. The PIF value for intermediate information processing rates is calculated from a linear interpolation between the lower and upper thresholds. A higher PIF value represents a greater operator information load. The information load threshold values can be adjusted to represent the operator's information processing capability.

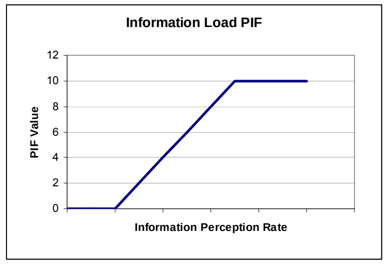


Figure 15, "Information Load PIF"

`info_upper_threshold:` Format: Integer. Range > ***`info_lower_threshold`***. In conjunction with the lower information loading threshold and the operator's average information processing rate, this parameter is used to calculate the information load PIF value. See "***`info_lower_threshold`***" for additional information.

"**alarm_update_time**": Format: double, double, double. Range: all values > 0.0. These parameters establish the recency criteria used by the operator when the use of memorized information is enabled. A three parameter Weibull distribution is used to describe the probability distribution for the use of old alarm state information. The Weibull distribution is given by equation (1):

$$F(t)=1-\exp\left[-\left(\frac{t-u}{\alpha}\right)^{\beta}\right]$$

(Equation 1)

Where:

double_1 = u (minimum time), seconds
double_2 = α parameter
double_3 = β parameter

When the use of memorized information is enabled, the operator will check to determine if the alarm state required to evaluate a procedural expectation of knowledge-based action prerequisite has been previously perceived. If the alarm state has been perceived, a Monte Carlo simulation is used to calculate an alarm update time from equation (1). If the age of the perceived alarm state is less than the alarm update time obtained from the Monte Carlo simulation, the operator will use the memorized information. If the age of the perceived alarm state is greater than the alarm update time, the operator will obtain the current alarm state from the control panel. This parameter can be used to prevent the operator for utilizing unacceptably old information.

"**component_update_time**": Format: double, double, double. Range: all values > 0.0. These parameters establish the recency criteria used by the operator when the use of memorized information is enabled. A three parameter Weibull distribution is used to describe the probability distribution for the use of old component state information. The Weibull distribution is given by equation (1). When the use of memorized information is enabled, the operator will check to determine if the component state required to evaluate a procedural expectation of knowledge-based action prerequisite has been perceived by the operator. If the component state has been perceived, a Monte Carlo simulation is used to calculate a component state update time from equation (1). If the age of the perceived alarm state is less than the component state update time obtained from the Monte Carlo simulation, the operator will use the memorized information. If the age of the perceived component state is greater than the alarm update time, the operator will obtain the current component state from the control panel. This parameter can be used to prevent the operator for utilizing unacceptably old information.

"parameter_update_time": Format: double, double, double. Range: all values > 0.0. These parameters establish the recency criteria used by the operator when the use of memorized information is enabled. A three parameter Weibull distribution is used to describe the probability distribution for the use of old parameter value information. The Weibull distribution is given by equation (1). When the use of memorized information is enabled, the operator will check to determine if the parameter required to evaluate a procedural expectation of knowledge-based action prerequisite has been perceived by the operator. If the parameter has been perceived, a Monte Carlo simulation is used to calculate a parameter update time from equation (1). If the age of the perceived parameter is less than the parameter update time obtained from the Monte Carlo simulation, the operator will use the memorized information. If the age of the perceived parameter is greater than the parameter update time, the operator will obtain the current parameter value from the control panel. This parameter can be used to prevent the operator for utilizing unacceptably old information.

skip_action_threshold: Format: double. Range: 0.0 – 1.0. This skip action threshold sets the minimum probability for generating a branching point for skipping a procedural step. Procedural steps in ADS-IDAC have three main components: (1) initial action activity, (2) expectations associated with the initial action activity, and (3) a non-response action that is executed if the action expectations are not met. The operator may skip either the initial action activity or the non-response action (evaluation of the action expectations cannot be skipped). The probability of skipping the initial action or non-response action is dynamically calculated based upon the baseline skip probability for the step component (specified in the procedure step input file), the type of procedure being followed, the step objectives, the relevance of the action to the operator's situational assessment, and certain PIFs. If the calculated skip probability for the initial action activity exceeds the skip action threshold, a branching point with two branches is generated. The procedure step initial action activity and associated expectation evaluation are skipped for skipped step branching path, while the step is executed on the complimentary branching path. The branching probability for the step skipping branch is set equal to the calculated branch probability and the branch probability for execution of the step is equal to the complement of the skip probability. If the calculated skip probability is less than the skip action threshold, the associated action is executed and no branching point is generated. The analyst can reduce the excessive generation of procedure step skipping branches by setting the skip action threshold to a higher value. Setting this parameter equal to 1.0 will prevent the operator from skipping procedural actions (i.e., all procedure initial action activities are executed).

`skip_non_response_threshold`: Format: double. Range: 0.0 – 1.0. This skip non-response action threshold sets the minimum probability for generating a branching point for skipping a non-response action in a procedure step. Procedural steps in ADS-IDAC have three main components: (1) initial action activity, (2) expectations associated with the initial action activity, and (3) a non-response action that is executed if the action expectations are not met. The operator may skip either the initial action activity or the non-response action (evaluation of the action expectations cannot be skipped). The probability of skipping the initial action or non-response action is dynamically calculated based upon the baseline skip probability for the step component (specified in the procedure step input file), the type of procedure being followed, the step objectives, the relevance of the action to the operator's situational assessment, and certain PIFs. If the calculated skip probability for the non-response action exceeds the skip non-response action threshold, a branching point with two branches is generated. The procedure step non-response action is skipped (if the initial action activity expectations are not met) on the skipped step branching path, while the non-response action is executed on the complimentary branching path. The branching probability for the non-response action skipping branch is set equal to the calculated branch probability and the branch probability for execution of the non-response action is equal to the complement of the skip probability. If the calculated skip probability is less than the skip non-response action threshold, the associated non-response action is executed (if the initial action activity expectations are not met) and no branching point is generated. The analyst can reduce the excessive generation of procedure step skipping branches by setting the skip non-response action threshold to a higher value. Setting this parameter equal to 1.0 will prevent the operator from skipping non-response actions (i.e., all procedure initial action activities are executed).

abnormal_threshold: Format: double. Range: 0.0 – 1.0. This parameter establishes the diagnostic threshold for an abnormal condition. The ADS-IDAC model includes a fuzzy logic diagnostic engine that supports the operator's situational assessment of the plant state. The parameters used in the diagnostic engine are described in the "KB_OAT(ODM)_Event_Matrix.txt" input file. Three major classes of plant events are included in the diagnostic process: (1) normal operating events, (2) anticipated operational occurrences, and (3) design basis accidents. Based on the information perceived by the operator, the diagnostic engine calculates a membership value for each possible diagnostic event. These membership values range from 0.0 to 1.0 and represent the degree of matching between the event symptoms and the symptoms that have been perceived by the operator. Each time step, ADS-IDAC calculates the maximum membership value of all events within the anticipated operational occurrence and design basis event categories. If this maximum membership value exceeds the abnormal signal threshold, the operator will conclude that an abnormal condition exists (depending on the current high level operator goal state). The abnormal condition diagnosis is not reset once an abnormal condition has been detected. This parameter effectively establishes the operator's sensitivity to detecting an abnormal event. If the parameter is set to a high value, the operator will require more information to support an abnormal condition diagnosis. A lower value reduces the information requirements for detecting an abnormal condition, but might result in the operator reaching a "false positive" conclusion for an abnormal event. Following the diagnosis of an abnormal event, the operator will suspend the execution of all low priority mental procedures. The diagnosis of an abnormal condition also influences the goal selection process.

priority_threshold: Format: Integer. Range: > 0. Following the identification of an abnormal condition (based on the abnormal signal threshold and diagnostic engine) or a reactor trip, the operator will suspend all low priority mental procedures. The priority of a mental procedure is specified in the "HardWiredDiagnosis.txt" input file and is represented by an integer value of 1 or greater. High priority procedures are associated with a low priority value (i.e., the highest priority procedures have a priority value of "1"). Once an abnormal condition has been detected, the operator suspends the execution of all mental procedures will a lower priority level than the specified threshold value (i.e., procedures with a priority value higher than the threshold). For example, if the priority threshold is set at 2, all mental procedures with a priority value of 3 or greater will be suspended. The purpose of this parameter is to allow the operator to interrupt mental procedures that are no longer appropriate following a reactor trip or during an accident event. Because the abnormal condition diagnosis is not reset, the suspension of low priority mental procedures can occur only once during an accident sequence. Thus, low priority mental procedures that are activated following the initial diagnosis of a reactor trip[9] or abnormal event are not automatically suspended and may be executed.

[9] The diagnosis of a reactor trip is controlled by the activation of the mental belief "Reactor_Trip" in the HardWiredDiagnsois.txt input file.

communication_time: Format: double. Range: > 0.0. Certain crew activities require coordination and communication between the control room operators. For example, only the Decision Maker can direct the performance of proceduralized actions and only the Action Taker can manipulate the control panel. Therefore, the execution of a procedure step requires the Decision Maker to direct the Action Taker to perform the specified action followed by a report from the Action Taker to the Decision Maker that the action had been accomplished. The time required to perform inter-crew communication is established by the nominal communication time parameter. The parameter establishes the communication delay time (in seconds) and is controlled by the sender of the information.

troubleshooting_branch_probability: Format: double. Range: 0.0 – 1.0. The troubleshooting probability sets the branching probability for activating the troubleshooting goal. If the value is greater than 0.999999, the operator will always activate the troubleshooting goal if the normal operation goal[10] is no longer appropriate. If the value is less than 0.000001, the operator will bypass the troubleshooting goal and always activate the monitoring goal when the normal operation goal is no longer appropriate. For intermediate values, a branching point will be generated with two operator goal branches. One branch will activate the troubleshooting goal with a branch probability equal to the troubleshooting probability. The other branch will activate the monitoring goal with a complimentary branching probability. When the troubleshooting goal is activated, the crew can implement knowledge-based actions to address the abnormal condition. This parameter is currently implemented only for the Decision Maker. A dummy value should be entered for the Action Taker.

procedure_use_branch_probability: Format: double. Range: 0.0 – 1.0. The procedure use probability sets the branching probability for enabling a transition from the troubleshooting goal to the "maintain global safety" goal upon the identification of a reactor trip condition. If this value is greater than 0.999999, the operator will always transition from the troubleshooting goal to the "maintain global safety" goal following the identification of a reactor trip condition. If this value is less than 0.000001, the goal transition from troubleshooting to "maintain global safety" margin is blocked. For intermediate values, a branching point will be generated with two operator goal branches. One branch will activate the transition from the troubleshooting goal to the "maintain global safety margin" goal with a branch probability equal to the troubleshooting probability. The other branch will block the transition to the "maintain global safety margin" goal. Effectively, this parameter allows the crew to transition from a knowledge-based approach to accident mitigation to a procedure following approach. This parameter is currently implemented only for the Decision Maker. A dummy value should be entered for the Action Taker.

[10] The normal operation goal is activated when the mental belief "Normal_Operation" in the HardWiredDiagnosis.txt input is activated.

4. Sample Input

```
Action_Taker 16
action_time_multiplier              1.0
confidence_level_for_acting_HWKB    0.80
use_memorized_info                  0.0
initial_scan_queue_limit            30
lower_info_load_threshold           0.0
upper_info_load_threshold           250.0
alarm_update_time                   100.0 1.0   1.0
component_update_time               100.0 1.0   1.0
parameter_update_time               100.0 1.0   1.0
skip_action_threshold               1.0
skip_non_response_threshold         1.0
abnormal_signal_threshold           0.4
mental_proc_priority_threshold      2
nominal_communication_time          1.0
troubleshooting_probability         0.0
procedure_use_probability           0.5
```

1. Purpose

The purpose of the "KB_AOT_Bias_Factors.txt" is to allow the analyst to simulate failed control panel instrumentation. In general, when an operator perceives the value of a plant indicator, the actual value of the parameter is obtained directly from the RELAP thermal-hydraulic model. In order to simulate a failed indicator, it is necessary to apply a bias factor to the RELAP generated data in order to model an instrument failure. Several indicator failure options are available, including additive errors, proportional errors, stuck instrumentation, and instruments that cannot read above or below a set threshold. When a parameter is biased in this manner, the operator may use inaccurate data when assessing the procedure step expectations, knowledge-based action prerequisites, and activation criteria for hard wired mental beliefs.

2. Input File Format

```
Number_of_biased_parameters          number_parameters
parameter_name    activation_time    option_code       bias_factor
.
.
.
```

3. Input Description

number_parameters: Format: Integer. Range ≥ 0. Parameter used to set the number of biased parameters that are included in the input file. If the value is set to 0, the reminder of the input file is ignored. If the value is greater than 0, an input data line (consisting of a string, double_1, integer, and double_2) must exist for each biased parameter. For example, if the number of biased parameters is equal to five, at least five data input lines must be supplied or an error will be generated during input file processing.

parameter_name – Format: string. Contains the name of the parameter to be biased. The parameter name must match a control panel parameter listed in the "ControlPanel.txt" input file. Spaces must not be used in the ***parameter_name*** (the underbar character "_" may be substituted for a space when needed).

activation_time – Format: Double. Range: > 0.0. Specifies the activation time for parameter bias factor. If the simulation time is less than the activation time, the parameter is unbiased and the operator will perceive the actual parameter value when the component is read from the control panel. If the simulation time is greater than the activation time, the parameter will be biased by the specified bias factor. The activation time must be greater than or equal to 0.0.

option_code – Format: Integer. Range: 3006, 3007, 3008, 3009, 3010, 3111, or 3112. Specifies the type of parameter bias to be applied to the perceived data. An integer code is used to specify the bias type. The following biasing categories are currently supported:

3006 (Integer code for "VGT"): The perceived parameter value will always be greater than the value specified by the ***bias_factor***. If the actual parameter value is greater than the ***bias_factor***, the actual parameter value will be perceived by the operator. If the actual parameter value is less than the ***bias_factor***, the operator will perceive the parameter as equal to the ***bias_factor***. This option can be used to model an indicator that will not read below a threshold value.

3007 (Integer code for "VGE"): The perceived parameter value will always be greater than or equal to the value specified by the ***bias_factor***. If the actual parameter value is greater than or equal to the ***bias_factor***, the actual parameter value will be perceived by the operator. If the actual parameter value is less than the ***bias_factor***, the operator will perceive the parameter as equal to the ***bias_factor***. This option has a similar effect as the "3006" option.

3008 (Integer code for "VEQ"): The operator will always perceive the parameter as equal to the ***bias_factor***. This option can be used to model an indicator that is stuck at a constant value.

3009 (Integer code for "VLE"): The perceived parameter value will always be less than or equal to the value specified by the ***bias_factor***. If the actual parameter value is less than or equal to the ***bias_factor***, the actual parameter value will be perceived by the operator. If the actual parameter value is greater than the ***bias_factor***, the operator will perceive the parameter as equal to the ***bias_factor***. This option can be used to model an indicator that will not read above a threshold value.

3010 (Integer code for "VLT"): The perceived parameter value will always be less than the value specified by the ***bias_factor***. If the actual parameter value is less than the ***bias_factor***, the actual parameter value will be perceived by the operator. If the actual parameter value is greater than the ***bias_factor***, the operator will perceive the parameter as equal to the ***bias_factor***. This option has a similar effect as the "3009" option.

3111 (Integer code for "VPROPORTIONAL"): The perceived parameter is equal to the actual parameter reading multiplied by the ***bias_factor***. For example, to model an indicator that consistently reads 10% too high, the ***bias_factor*** would be set to 1.10. This option can be used to model an indicator that has a constant proportional error.

3112 (Integer code for "VADDITIVE"): The perceived parameter value is equal to sum of the actual parameter value and the value of the ***bias_factor***. This option can be used to model an indicator that has a constant offset bias.

bias_factor **–** Format: Double. Range: Consistent with indicator output. Specifies the bias factor that will be applied to the perceived parameter based on the bias option specified by the analyst. The analyst should ensure that the bias factor value and units are consistent with the normal output range for the associated indicator.

Input data lines are repeated until the bias factors for each desired parameter are specified.

4. Sample Input

```
Number_of_biased_parameters   2
SG_A_WR_Level     0.0   3007   0.16
SG_C_WR_Level     100.0 3112   0.15
```

The above example specifies two types on bias factors. In the case of the "SG_A_WR_Level" parameter, beginning at the start of the simulation (activation time equals 0.0 seconds), the SG A wide range level indicator will read no lower than 0.16 (16%). Thus, when the parameter is greater than 0.16, the actual value will be perceived, but when the actual value is less than 0.16, the operator will perceive a value of 0.16. For the case of "SG_C_WR_Level", a constant bias offset of 0.15 is added to the actual parameter value when the simulation time exceeds 100.0 seconds (prior to 100.0 seconds, no bias factor is applied to the indicator).

1. Purpose

Diagnosis actions are used to implement the knowledge-based problem solving approach when the control room crew is implementing the "troubleshooting" goal. Diagnostic actions are grouped within functional areas that are aligned with the imbalance events described in the "KB_OAT_Event_Matrix.txt" file. Because many possible actions may be available to address a specific imbalance event, each diagnosis action is assigned a priority level and a set of prerequisite conditions. Diagnosis actions can only be implemented if the specified prerequisite conditions with higher priority actions are executed prior to low priority actions. In the current version of ADS-IDAC, all knowledge-based actions are initiated by the Decision Maker; therefore the diagnosis action file for the Action Taker does not affect the simulation. However, the ADS-IDAC has been written to allow for a later upgrade to implement Action Taker initiated knowledge-based actions. Consequently, the input file "KB_OAT_Diagnosis_Actions.txt" must be included in the code input. Until the initiation of knowledge-based actions is fully implemented for the Action Taker, it is recommended that no diagnoses actions by specified in this input file. Diagnosis actions have been fully implemented for the Decision Maker and may included in the "KB_ODM_Diagnosis_Actions.txt" input file.

2. Input File Format

```
number_of_diagnoses            number_of_diagnoses
```

3. Input Description

number_of_diagnoses: Format: Integer. Range: ≥ 0. This parameter specifies the number of diagnosis event categories used for knowledge-based actions. This parameter must equal the actual number of symptoms entered in this file. Diagnosis actions have not been fully implemented for the Action Taker. Therefore, the input parameters in this file do not impact the simulation. However, an input error will be generated if this file is not present. Therefore, for the Action Taker, it is recommended to set this parameter to 0.

See "KB_ODM_Diagnosis_Actions.txt" for additional information on specifying knowledge-based diagnostic actions.

4. Sample Input

```
number_of_diagnoses            0
```

1. Purpose

The purpose of input file KB_OAT_Event_Matrix.txt is to specify the symptoms and events that are used to generate diagnostic results. The input file consists of a listing of symptoms used to support event diagnosis and a relationship value matrix that provides the linkage between symptoms and events. Four general types of event types are used in ADS_IDAC: normal operating events ("normal"), anticipated operational occurrences ("AOO"), design basis accidents ("DBA"), and mass, energy, or momentum flow imbalances ("Imbalance"). Normal operating events refer to conditions normally experienced during power operation (e.g., load changes) and instrument failures. Anticipated operational occurrences are abnormal events that are expected to occur during the life span of the facility but are not expected to cause fuel damage. Design basis accidents are more serious events that are not normally expected to occur, but may result in some fuel damage. Imbalance events are used to support knowledge-based problem solving by identifying high level plant functions that are not in an equilibrium state. Because the diagnostic engine uses a fuzzy logic inference approach, the relationship values that link symptoms and events should be viewed as set membership values rather than a strict confidence level or subjective probability.

2. Input File Format

```
Number_of_Symptoms          num_symptoms
symptom_1
symptom_2
.
.
.
symptom_n

Number_of_Events            num_events
event_name_1      type_1      value_11     value_12     ...
      value_1n
event_name_2      type_2      value_21     value_22     ...
      value_2n
.           .           .           .                 .
.           .           .           .                 .
.           .           .           .                 .
event_name_m      type_m      value_m1     value_m2     ...
      value_mn
```

3. Input Description

num_symptoms: Format: Integer. Range: $\geq$ 0. This parameter specifies the number of event symptoms used in the fuzzy logic diagnostic engine. This parameter must equal the actual number of symptoms entered in this file.

symptom_i: Format: String. Range: not applicable. This parameter specifies the symptom name used to support the diagnostic engine. Enter only one symptom name per line and ensure the number of symptoms equals ***num_symptoms*** or an error will be generated during input file processing. The symptom name must exactly match a mental belief entered into the "HardWired_Diagnosis.txt" input file. The symptom name must not contain the space character (use the underbar character ("_") rather than a space when needed).

num_events: Format: Integer. Range: ≥ 0. This parameter specifies the number of events that are included in the diagnostic process. This parameter must equal the number of event data rows entered in this input file. Each event data row consists of the event name (***event_i***), the event type (***type_i***), and a set of relationship values (***value_ij***) that establishes the linkage between the event and each symptom. Each event data row must include exactly ***num_symptoms*** relationship values.

event_name_i: Format: String. Range: any. This parameter specifies the event name. Spaces must not be used (use the underbar character ("_") rather than a space when needed. For imbalance events, an event data row must be entered for each functional item included in the "KB_OAT_System_Decomposition.txt" input file or an error message might be generated during the simulation run.

type_i: Format: String. Range: Entry must be one of the following types: "Normal", "AOO", "DBA", or "Imbalance". This parameter specifies the event category. Only events included within the "AOO" or "DBA" event type are used by the operator to identify when an abnormal condition has occurred. Imbalance events are used to support knowledge-based problem solving and are used to identify non-equilibrium conditions in mass, energy, or momentum flow.

value_ij: Format: Double. Range: 0.0 – 0.99. This parameter provides the relationship for *event_i* and *symptom_j*. A value of 0.0 indicates that the event and associated symptom are not related (i.e., the symptom and event occur independently). An increasing value indicates an increasingly strong relationship between *event_i* and *symptom_j*. In general, a higher relationship value indicates that operator has a greater level of confidence that the symptom would be observed given that the associated event has occurred.

Because relationship values are intended to represent the operator's mental model of plant behavior, numerical relationship values can be assigned using heuristic rules rather than formal thermal-hydraulic or probability analysis. The following heuristic rules have been used to grouped event symptoms into the broad categories of primary, secondary, and tertiary symptoms:

- Primary symptoms directly relate to the initiating event and are expected to be observed with a high degree of confidence;

- Secondary symptoms are the result of the primary symptoms and are normally expected to be observed, but with a lower degree of confidence than primary symptoms; and
- Tertiary symptoms may arise due to the presence of primary or secondary symptoms but can be mitigated by either control system operation or thermal hydraulic feedback mechanisms. Consequently, tertiary symptoms may not be observed and are assigned a low degree of confidence.

Relationship values were assigned based on engineering judgment and the guidelines of Table 3.

Table 3. Symptom-Event Relationship Values

Symptom Type	Relationship Value Range[(1)]
Primary	0.7 – 0.99
Secondary	0.4 – 0.7
Tertiary	0.1 – 0.4

[(1)] A higher value indicates a stronger relationship between the symptom and event

4. Sample Input

```
Number_of_Symptoms      5
Power_Decrease
Tave_Decrease
Pressurizer_Level_Decrease
RCS_Pressure_Decrease
SG_A_Steam_Flow_Increase

Number_of_Events  4
PZR_Level_Failure_(Low)        Normal       0.0    0.0    0.9    0.5
      0.0
Reactor_Trip                   AOO          0.9    0.6    0.5    0.4
      0.0
LOCA_Inside_Containment        DBA          0.0    0.0    0.9    0.6
      0.0
Energy_Imbalance_PZR_Low       Imbalance    0.0    0.0    0.0    0.7
      0.0
```

This sample input file identifies five symptoms that will be used by the diagnostic process (Power_Decrease, Tave_Decrease, Pressurizer_Level_Decrease, RCS_Pressure_Decrease, and SG_A_Steam_Flow_Increase). Each of the symptoms should have a corresponding mental belief data entry (with an identical mental belief name) in the operators "KB_OAT_HardWired_Diagnosis.txt" file. Four events have been specified: (1) a pressurizer level control failure during normal operation (PZR_Level_Failure_(Low)), (2) a reactor trip condition (Reactor_Trip), (3) a loss of coolant design basis accident

(LOCA_Inside_Containment), and (4) and imbalance category event associated with a low energy condition in the pressurizer (Energy_Imbalance_PZR_Low). The relationships between symptoms and events can be better seen in Table 4. In this case, the pressurizer level control failure is strongly associated with a decreasing pressurizer level decrease (0.9 relationship value) and less strongly associated with a decreasing RCS pressure (0.5 relationship value). Similarly, the reactor trip event is strongly associated with a power decrease and less strongly linked to decreasing average temperature, decreasing pressurizer level, and decreasing RCS pressure. In this example, none of the events are associated with an increase in steam flow in SG A.

Event	Symptoms				
	Power Decrease	Tave Decrease	Pressurizer Level Decrease	RCS Pressure Decrease	SG-A Steam Flow Increase
PZR Level Failure	0.0	0.0	0.9	0.5	0.0
Reactor Trip	0.9	0.6	0.5	0.4	0.0
LOCA Inside Containment	0.0	0.0	0.9	0.6	0.0
Energy Imbalance PZR Low	0.0	0.0	0.0	0.7	0.0

Table 4: Sample Relationship Table

1. Purpose

The input file "KB_OAT_HardWired_Diagnosis.txt" establishes the mental beliefs that may be activated by the operator during the simulation. Once a mental belief is activated, the operator may use the mental belief to initiate a mental (memorized) procedure, activate other mental beliefs, or for the evaluation of procedure step expectations and knowledge-based action prerequisites. The ADS-IDAC code also utilizes the mental beliefs "Normal_Operation" and "Reactor_Tripped" (if they exist) for the goal selection process. Mental beliefs are activated based on satisfying the specified prerequisite alarm state(s), component state(s), parameter state(s), control value state(s), mental belief state(s), and procedure activation(s). The confidence level for a mental belief calculated from the ratio of satisfied prerequisite states to the total number of prerequisites specified for the mental belief.

A mental belief is activated when the following conditions have been met: (1) the mental belief confidence is greater than the specified activation confidence, (2) mental belief activation is not blocked by the reset time delay, and (3) the mental belief has been enabled by specifying a branching probability greater than 0.000001. The hard wired diagnosis branch probability controls the generation of a mental belief activation branch. If the branch probability is less than 0.000001, the mental belief will not be activated. If the branch probability is greater than 0.999999, a single mental belief activation branch will be generated if the confidence level exceeds the activation threshold and activation is not blocked by the reset timer. For intermediate branch probability values, two mental belief branches will be activated, one branch which enables mental belief activation and one branch that bypasses (blocks) mental belief activation. Once a mental belief has been activated, the operator will initiate the associated mental procedure (if specified) after the activation time delay has elapsed. The reset time delay blocks re-activation of the mental belief until the reset time has elapsed. Once the reset time delay has elapsed, the mental belief (and the associated mental procedure) can be re-activated provided the required confidence level has been reached.

A subset of mental beliefs is also used to support the fuzzy logic diagnosis engine. Each symptom included in the "KB_OAT_Event_Matrix.txt" input file must have a companion mental belief in the "KB_OAT_HardWired_Diagnosis.txt" input file with the same name (i.e., the event matrix symptom name is identical to a hard wired diagnosis mental belief name). To avoid the generation of unnecessary branches associated with diagnosis symptoms, the branch probability for symptom related mental beliefs may be set to 0.0 and the activation confidence set to 1.0 (the diagnosis engine directly reads the mental belief confidence – no branch generation is necessary).

2. Input File Format

```
Number_of_Hardwired_Diagnosis      number_of_items

count        mental_belief_name
activation_confidence              activation_confidence
branch_probability                 branch_probability
activation_delay_time              double_1    double_2
      double_3
reset_delay_time                   double_1    double_2
      double_3

Number_of_expected_alarm_state     number_of_alarms
alarm_name_1            alarm_state_1
.                       .
.                       .
alarm_name_n            alarm_state_n

Number_of_expected_component_state number_of_components
component_name_1        component_state_1
.                       .
.                       .
component_name_n        component_state_n

Number_of_expected_parameter_value number_of_parameters
parameter_name_1        logical_operator  value_1    value_2
.                       .                       .          .
.                       .                       .          .
parameter_name_n        logical_operator  value_1    value_2

Number_of_manipulative_control     number_of_controls
control_name_1          minimum_control_value_1
.                       .
.                       .
control_name_n          minimum_control_value_n

Number_of_mental_belief            number_of_mental_beliefs
mental_belief_name_1    mental_belief_state_1
.                       .
.                       .
mental_belief_name_n    mental_belief_state_n

Number_of_procedure_activity       number_of_procedures
procedure_name_1        procedure_state_1
.                       .
.                       .
procedure_name_n        procedure_state_n

Mental_procedure_priority          priority
procedure_name          step_name


.
.
.
```

3. Input Description

number_of_items: Format: Integer. Range: ≥ 0. Specifies the number of mental beliefs described in the input file.

count: Format: Integer. Range > 0. This parameter is not used by the ADS-IDAC but is included for the convenience of the analyst to verify the number of mental beliefs specified by the ***number_of_items*** variable.

mental_belief_name: Format: String. Range: any. Descriptive name for the mental belief. ADS-IDAC recognizes two special mental beliefs: "Normal_Operation" and "Reactor_Tripped". If these special mental beliefs are present, they are used to support the operator goal selection process and suspension of low priority mental procedures following a reactor shutdown. Spaces must not be used (use the underbar character ("_") rather than a space when needed.

activation_confidence: Format: Double. Range: ≥ 0.0. Specifies the minimum confidence level necessary to enable mental belief activation and branch generation. The mental belief confidence is equal to the ratio of satisfied conditions to the total number of prerequisite conditions. The activation confidence level influences the activation logic for the associated mental belief. By setting the activation confidence to a low value (e.g., 0.05), the mental belief can be activated when a small percentage of the prerequisite conditions are satisfied. A high activation confidence level (e.g., 0.95) would require that all (or nearly all) of the prerequisite conditions are satisfied. Therefore, the activation confidence level can be adjusted to shift mental belief activation from "or" gate logic to "and" gate logic. Intermediate values of the activation confidence can be used to model "k of n" logic. For example, if a mental belief has five prerequisite conditions and satisfying any three conditions is sufficient for activation, setting the activation confidence in the range of $0.4 < x < 0.8$ would simulate 3 out 5 logic.

Setting the activation_confidence to 1.0 should be avoided since the numerical methods used to calculate the confidence level may preclude activation of the mental belief even if all prerequisite conditions are satisfied. In practice, the calculated mental belief confidence is normally in the range of 0.0 to slightly less than 1.0 (a small bias value of 0.00001 is added to the total number of prerequisites to avoid a "division by zero" error if no prerequisite conditions are specified for the mental belief.

branch_probability: Format: double. Range: 0.0 – 1.0. This parameter sets the branching probability for activating the mental belief. If the value is greater than 0.999999, a single branch will be generated to activate the mental belief provided the minimum confidence level is met and the reset time delay has not blocked activation. If the value is less than 0.000001, the operator will bypass/block mental belief activation and the associated mental procedure (if supplied) will not be initiated. For intermediate values, a branching point will be generated with two mental belief branches. One branch will activate the mental belief (and initiate the associated mental procedure if applicable) with a branch probability equal to the branch probability. The other branch will bypass (block) mental belief activation with a complimentary branching probability. A bypassed mental belief may become reactivated once the reset time delay has elapsed (provided the appropriate prerequisite conditions are met).

***"activation_delay_time"*:** Format: Double, Double, Double. Range: all values ≥ 0.0. These parameters establish the activation delay time for initiation of the associated mental procedure (if supplied). Upon activation of a mental belief, the associated mental procedure will be added to the operator's procedure queue. However, a mental procedure will not be initiated until the activation delay time has elapsed. This parameter allows the analyst to separate (in time) the activation of the mental belief and the execution of the associated mental procedure. A three parameter Weibull distribution is used to describe the probability distribution for the use of activation time delay. The Weibull distribution is given by equation (2):

$$F(t) = 1 - \exp\left[-\left(\frac{t-u}{\alpha}\right)^{\beta}\right]$$

(Equation 2)

Where:

double_1 = u (minimum time), seconds
double_2 = α parameter
double_3 = β parameter

Upon activation of a mental belief , a Monte Carlo simulation is used to calculate the mental procedure activation time delay using equation (2). Once the activation time delay has elapsed, the mental procedure can be initiated by the operator.

"reset_delay_time"*:** Format: Double, Double, Double. Range: all values ≥ 0.0. These parameters establish the reset time delay for a mental belief. The purpose of the reset time delay is to allow the analyst to create mental beliefs that can be activated multiple times during a simulation run. The reset time delay provides a dormancy time during which the mental belief cannot be reactivated (even if the necessary prerequisite conditions are met). This provides more realistic control over certain repetitive operator actions such as control of auxiliary feed water flow rate. Similar to the "activation_delay_time***", a three parameter Weibull distribution is used to describe the probability distribution for the use of old component state information. The Weibull distribution is given by equation (2).

Mental Belief Prerequisite Conditions

1. ***Alarm States***

number_of_alarms: Format: Integer. Range: ≥ 0. Specifies the number of alarm state prerequisites for the associated mental belief. If ***number_of_alarms*** is set to 0, no further alarm state data should be entered. If ***number_of_alarms*** is greater than or equal to 1, input data for each alarm (i.e., ***alarm_name*** and ***alarm_state***) must be supplied or an error will be generated during input file processing.

alarm_name_i: Format: String. Range: any. Parameter identifies the alarm. The ***alarm_name*** must match an alarm specified in the "ControlPanel.txt" input file.

alarm_state_i: Format: Integer. Range: 3022, 3023. Specifies the alarm activation state. The alarm state should be specified as either 3022 (integer code "VON" for alarm activation) or 3023 (integer code "VOFF" for alarm cleared).

2. ***Component States***

number_of_components: Format: Integer. Range: ≥ 0. Specifies the number of component state prerequisites for the associated mental belief. If ***number_of_ components*** is set to 0, no further component state data should be entered. If ***number_of_ components*** is greater than or equal to 1, input data for each component (i.e., ***component_name*** and ***component_state***) must be supplied or an error will be generated during input file processing.

component_name_i: Format: String. Range: any. Parameter identifies the component. The ***component_name*** must match a component specified in the "ControlPanel.txt" input file.

component_state_i: Format: Integer. Range: 3022, 3023. Specifies the component activation state. The component state should be specified as either 3022 (integer code "VON" for component is activated) or 3023 (integer code "VOFF" for component stopped or off).

3. ***Parameter States***

number_of_parameters: Format: Integer. Range: ≥ 0. Specifies the number of parameter state prerequisites for the associated mental belief. If ***number_of_parameters*** is set to 0, no further parameter state data should be entered. If ***number_of_parameters*** is greater than or equal to 1, input data for each parameter state (i.e., ***parameter_name, logical_operator, value_1*** and

value_2) must be supplied or an error will be generated during input file processing.

parameter_name_i: Format: String. Range: any. Identifies the control panel indicator associated with the prerequisite condition. The ***parameter_name*** must match a parameter specified in the "ControlPanel.txt" input file.

logical_operator: Format: Integer. Range: allowable values are 3006, 3007, 3008, 3009, 3010, and 3032. The ***logical_operator*** integer code value is used to specify the following conditions:

- 3006 (Integer code for "VGT") - Condition satisfied when parameter > ***value_1***
- 3007 (Integer code for "VGE") - Condition satisfied when parameter ≥ ***value_1***
- 3008 (Integer code for "VEQ") – Condition satisfied when parameter = ***value_1***
- 3009 (Integer code for "VLE") - Condition satisfied when parameter < ***value_1***
- 3010 (Integer code for "VLT") - Condition satisfied when parameter ≤ ***value_1***
- 3032 (Integer code for "VBETWEEN") – Condition satisfied when
  value_1 < parameter < ***value_2***

value_1: Format: double. Range: any. Specifies the parameter threshold value for condition activation. The entered value should be consistent with the indicator range. For logical operator 3032 ("VBETWEEN") ***value_1*** must be the lower end of the threshold range.

value_2: Format: double. Range: any. Although this parameter must always be entered, it is only used for logical operator 3032 ("VBETWEEN"). For logical operator 3032 ("VBETWEEN") ***value_2*** must be the upper end of the threshold range. For all other logical operators, a dummy value should be entered (e.g., "0.0"). The entered value should be consistent with the indicator range.

4. *Control States*

number_of_controls: Format: Integer. Range: ≥ 0. Specifies the number of control state prerequisites for the associated mental belief. If ***number_of_ controls*** is set to 0, no further control state data should be entered. If ***number_of_controls*** is greater than or equal to 1, input data for each control (i.e., ***control_name*** and ***minimum_control_value***) must be supplied or an error will be generated during input file processing.

control_name_i: Format: String. Range: any. Identifies the control panel controller associated with the prerequisite condition. The ***control_name*** must match a controller specified in the "ControlPanel.txt" input file.

minimum_control_value_i: Format: Double. Range: any. Specifies the threshold control value necessary to satisfy the prerequisite condition. When the specified controller is adjusted to a control value equal to or greater than the ***minimum_control_value,*** the prerequisite condition is satisfied.

5. ***Mental Belief States***

number_of_mental_beliefs: Format: Integer. Range: $\geq$ 0. Specifies the number of mental belief state prerequisites for the associated mental belief. If ***number_of_mental_beliefs*** is set to 0, no further mental belief state data should be entered. If ***number_of_mental_beliefs*** is greater than or equal to 1, input data for each mental belief state (i.e., ***mental_belief_name*** and ***mental_belief_state***) must be supplied or an error will be generated during input file processing.

mental_belief_name_i: Format: String. Range: any. Identifies the mental belief associated with the prerequisite condition. The ***mental_belief_name*** must match a mental belief entry in the "KB_OAT_HardWired.txt" input file.

mental_belief_state_i: Format: Integer. Range: allowable values are 3020, 3021, and 3045. The ***mental_belief_state*** is specified using the following integer code values:

- 3020 (integer code for "VFAILED") Mental belief is not active
- 3021 (integer code for "VSUCCEED") Mental belief is active
- 3045 (integer code for "VNONE") No status information available or mental belief not active

6. ***Procedure States***

number_of_procedures: Format: Integer. Range: $\geq$ 0. Specifies the number of procedure state prerequisites for the associated mental belief. If ***number_of_procedures*** is set to 0, no further procedure state data should be entered. If ***number_of_procedures*** is greater than or equal to 1, input data for each procedure state (i.e., ***procedure_name*** and ***procedure_state***) must be supplied or an error will be generated during input file processing.

procedure_name_i: Format: String. Range: any. Identifies the procedure name associated with the prerequisite condition. The ***procedure_name*** must be consistent with a procedure listed in the "Procedures.txt" input file.

procedure_state_i: Format: Integer. Range: allowable values are 3045, 3089, 3097, 3126, and 3127. The ***procedure_state*** is specified using the following integer code values:

- 3045 (integer code for "VNONE") No status information available
- 3089 (integer code for "VINTERRUPT") Procedure abandoned
- 3097 (integer code for "VDONE") Procedure Complete
- 3126 (integer code for "VACTIVE") Procedure in use
- 3127 (integer code for "VPAUSE") Procedure temporarily suspended

priority: Format: Integer. Range: $\geq$ 1. Specifies the priority level of the associated mental procedure. If no mental procedure is associated with the mental belief, a dummy integer value should be entered. Lower priority numbers indicate a higher procedure priority. The priority is used to select mental procedures to be performed (higher priority procedures are performed before lower priority procedures) and block continued performance of low priority procedures following detection of an abnormal condition (see the ***priority_threshold*** discussion in the "ActionTaker.txt" input file description).

procedure_name: Format: String. Range: any. The ***procedure_name*** and ***step_name*** must be consistent with a procedure listed in the "Procedures.txt" input file. All mental procedures should include the prefix "MPBG_" to denote the associated procedure as activated by a mental belief. If no procedure is associated with the mental belief, the string "NONE" should be entered for the ***procedure_name***.

step_name: Format: String. Range: any. The ***procedure_name*** and ***step_name*** must be consistent with a procedure listed in the "Procedures.txt" input file. If no procedure is associated with the mental belief, the string "NONE" should be entered for the ***step_name***.

4. <u>Sample Input</u>

```
Number_of_Hardwired_Diagnosis   5

1 Tave_Increase
activation_confidence   0.0
branch_probability      0.0
activation_delay_time   0.0         1.0   1.0
reset_delay_time        3600.0      1.0   1.0
Number_of_expected_alarm_state      0
Number_of_expected_component_state  0
Number_of_expected_parameter_value  4
Tave-Tref         3007  1.0   0.0
RATE_Loop_A_Tave  3007  1.0   0.0
RATE_Loop_B_Tave  3007  1.0   0.0
RATE_Loop_C_Tave  3007  1.0   0.0
Number_of_manipulative_control      0
```

```
Number_of_mental_belief             0
Number_of_procedure_activity        0
Mental_procedure_priority     5
None        None

2 Align_B_MDAFP_Flow_Path
activation_confidence   0.8
branch_probability      1.0
activation_delay_time   50.0        20.0  2.0
reset_delay_time        10000.0     1.0   1.0
Number_of_expected_alarm_state      0
Number_of_expected_component_state  1
B_MDAFP_On              3022
Number_of_expected_parameter_value  0
Number_of_manipulative_control      0
Number_of_mental_belief             1
Align_A_MDAFP_Flow_Path       3045
Number_of_procedure_activity        0
Mental_procedure_priority           5
MPBG_Align_MDAFP_Valves       Step_1

3 Align_A_MDAFP_Flow_Path
activation_confidence   0.8
branch_probability      1.0
activation_delay_time   30.0        20.0  2.0
reset_delay_time        10000.0     1.0   1.0
Number_of_expected_alarm_state      0
Number_of_expected_component_state  1
A_MDAFP_On              3022
Number_of_expected_parameter_value  0
Number_of_manipulative_control      0
Number_of_mental_belief            1
Align_B_MDAFP_Flow_Path       3045
Number_of_procedure_activity        0
Mental_procedure_priority     1
MPBG_Align_MDAFP_Valves       Step_1

4 Normal_Operation
activation_confidence   0.8
branch_probability      0.0
activation_delay_time   0.0   1.0   1.0
reset_delay_time        10.0  1.0   1.0
Number_of_expected_alarm_state      0
Number_of_expected_component_state  0
Number_of_expected_parameter_value  3
Tave-Tref         3032  -1.5        1.5
PZR_Pressure      3032  2200.0      2300.0
PZR_Level         3032  0.40        0.55
Number_of_manipulative_control      0
Number_of_mental_belief             0
Number_of_procedure_activity        0
Mental_procedure_priority           5
NONE        NONE

5 Reactor_Tripped
activation_confidence   0.8
branch_probability      1.0
```

```
activation_delay_time   0.0         1.0   1.0
reset_delay_time        10000.0     1.0   1.0
Number_of_expected_alarm_state      1
A_Reactor_Trip    3022
Number_of_expected_component_state  0
Number_of_expected_parameter_value  0
Number_of_manipulative_control      0
Number_of_mental_belief             0
Number_of_procedure_activity        0
Mental_procedure_priority     5
NONE        NONE
```

____

The above example illustrates three categories of mental beliefs: (1) event symptom mental belief ("Tave_Increase"), (2) mental belief intended to activate a memorized procedure ("Align_A_MDAFP_Flow_Path" and "Align_B_MDAFP_Flow_Path"), and (3) use of the special ADS-IDAC recognized mental beliefs for normal operation and the reactor tripped plant state. These mental beliefs can be interpreted as follows:

- Tave_Increase: No branches are generated by the mental belief since the purpose of this entry is only to support the diagnostic engine (***branch_probability*** = 0.0). Dummy values have been entered for the activation time delay, the reset time delay, and the activation confidence since these values will have no impact on the simulation (since no branch will be generated by the mental belief). The mental belief confidence is based on four parameter values: (1) a reactor coolant system temperature deviation greater than or equal to 1.0 F, and (2) the rate of any loop average temperature greater than or equal to 1.0 F/minute. Since four prerequisite conditions are specified, the mental belief confidence can be any of five possible states depending on how many prerequisite conditions are met (e.g., 0.0, 0.25, 0.5, 0.75, and ~1.0). No mental procedure is associated with this mental belief.

- Align_A_MDAFP_Flow_Path: A single branch will be generated when this mental belief is activated (***branch_probabiltiy*** = 1.0). The mental belief is activated when the confidence level exceeds an ***activation_confidence*** of 0.8. Because only two prerequisite conditions are specified (one component state and one mental belief), a mental belief confidence level of 0.8 can only be reached if both prerequisite conditions are met. The prerequisite conditions allow the mental belief to be activated only if the operator perceives that the A train motor driven auxiliary feed water pump (A_MDAFP) is running and the operator has not activated the "Align_B_MDAFP_Flow_Path" mental belief (this condition prevents the associated mental procedure from being executed twice since both the "Align_A_MDAFP_Flow_Path" and "Align_B_MDAFP_Flow_Path" initiate the same mental procedure). Once the mental belief is activated, the operator will initiate mental procedure "MPBG_Align_MDAFP_Valves Step_1" after the activation time delay (approximately 30.0 seconds) has elapsed. The reset time delay of at least 10,000 seconds will preclude reactivation of the mental belief for any simulation lasting less than approximately 165 minutes.

- Normal_Operation: No branching is generated as a result of this mental belief since the ***branch_probability*** is set to 0.0. Therefore, dummy values are entered for the activation time delay, the reset delay time, and the associated mental procedure since these parameters have no impact of the simulation execution. However, the ***activation_confidence*** is used by ADS_IDAC in the goal selection process to diagnosis a normal operating state (Decision Maker only). If the mental belief confidence level is greater than the ***activation_confidence*** (in this case, 0.8), the selection of a normal operating high level goal is enabled. Three parameter states are used as the prerequisite conditions for this mental belief: (1) reactor coolant system temperature deviation, (2) pressurizer pressure, and (3) pressurizer level. Each of these parameter states is activated when the associated parameter falls between the bounds specified by ***value_1*** and ***value_2***. Since only three prerequisites are specified and the activation probability is 0.8, all three conditions must be met to enable the normal operating goal.

- Reactor_Tripped: A single branch is generated when the "Reactor_Tripped" mental belief is activated (***branch_probability*** is set equal to 1.0). If the branch_probability were set to an intermediate value (e.g., 0.5), then two branches would be activated if the mental belief confidence exceeded the ***activation_confidence*** (set to 0.8 in this case). The first branch would activate the mental belief while the second branch blocks the mental belief – in this manner the effect of the operator failing to activate the mental belief can be simulated. In this example only a single alarm condition is needed to activate the mental belief (i.e., activation of the reactor trip alarm). The reset probability is set to a high value (~165 minutes), effectively limiting the mental belief to a single activation during most simulation runs. Although no mental procedure is associated with the "Reactor_Tripped" mental belief, the activation of this belief is recognized by the ADS-IDAC code and performed two key functions: (1) supports the goal selection process and (2) blocks further execution of low priority mental belief procedures.

1. Purpose

Maneuvering actions enable ADS-IDAC to model continuous control manipulations without the need to activate mental beliefs or related procedures. Maneuvering actions are activated by the execution of a procedure step with the an action name beginning with the prefix "MANEUVER_". A maneuvering action models a simple control system that actuates a single component to maintain a target parameter within a specified band. For example, some plant emergency procedures require the operators to throttle emergency core cooling flow to maintain a specified sub-cooling margin in the reactor coolant system. A maneuvering action could be created that controlled the positioning of an emergency core cooling throttle valve as needed to maintain sub-cooling margin within a target band. Maneuvering actions allowed the ADS-IDAC operator model to exercise complex control manipulations without a significant amount of knowledge base development.

As the sophistication of the ADS-IDAC knowledge base increased, it was no longer necessary to rely on maneuvering actions to exercise these types of control manipulations. Instead, mental beliefs and mental procedures can now be used to exercise similar control outputs in a more realistic manner. Specifically, mental beliefs can be constructed to allow the operator to control more than a single component or consider more than a one target parameter and control band. Additionally, maneuvering control actions bypass certain aspects of the information gathering and perception filtering processes – the use of mental beliefs and mental procedures does not bypass these cognitive processes. Consequently, the use of maneuvering actions in ADS-IDAC is no longer recommended. Maneuvering action capabilities have been maintained as a legacy feature of ADS-IDAC, but use of this feature will not be described in this manual. Because some older ADS-IDAC project files may use maneuvering action capability, the program will attempt to read the "KB_OAT_Maneuvering_Action.txt" input file during program initiation. For new project problems, the "KB_OAT_Maneuvering_Action.txt" input file must be included in the input directory, but it is recommended that no maneuvering actions be specified.

2. Input File Format

`Number_of_operator_Maneuver_action` ***`number_of_maneuvering_actions`***

3. Input File Description

`number_of_maneuvering_actions:` Format: Integer. Range: ≥ 0. Specifies the number of maneuvering actions described in the input file. For new projects, it is recommended that this parameter be set to 0 and no maneuvering actions be specified.

4. Sample Input

```
Number_of_operator_Maneuver_action  0
```

____

This example simply specifies that no maneuver actions are specified for the project – this is the recommended content of the "KB_OAT_Maneuvering_Actions.txt" input file.

For legacy purposes only, the following example provides an example of a fully specified maneuvering action input file:

```
Number_of_operator_Maneuver_action  1

Maneuver_name                       MANEUVER_Throttle_HPI_to_50_SCM
Interactive_control_and_range       X_HPI   0.0    1.0
Targeted_parameter_name             SCM_113
LB_target_UB_of_the_parameter       45.0   50.0   55.0
Targeted_parameter_rate_name        RATE_SCM_113
Rate_boundary                       -1.0    1.0
Ratio_control_table_3_by_3
Change_rate_less_than_LB            50.0   0.0    0.0
Change_rate_in_between              25.0   0.0    -25.0
Change_rate_greater_than_UB         0.0    0.0    -50.0
Maneuver_time_interval              15.0
Maneuver_total_duration             3600.0
```

1. Purpose

The purpose of the "KB_OAT_Safety_Parameter.txt" input file is to support calculation of the criticality of system condition performance influencing factor (PIF). The criticality of system condition PIF represents the operator's perception of the level of degradation of key safety functions. This PIF is loosely based on the safety parameter display system used in U.S. nuclear plant control rooms. The value of the system criticality PIF corresponds to the aggregate deviation of key safety parameters from a nominal value. This input file identifies the parameters used to calculate this PIF, the threshold limits associated with each parameter, and the weighting factors used to aggregate the parameter contributions. Typical parameters used to calculate the system criticality PIF include reactor coolant system subcooling margin, wide range steam generator water levels, pressurizer water level, and reactor vessel water level. The contribution from each identified parameter to the overall criticality of system condition PIF value is denoted as the parameter criticality ($PIF_{Parameter\ Criticality}$). Given a set of high and low threshold limits, the parameter criticality corresponds to the magnitude of the parameter's deviation from a nominal "safe" condition (see Figure 16).

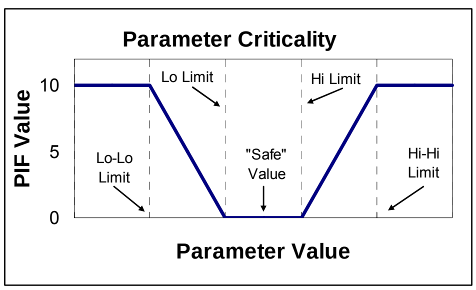


Figure 16, "Parameter Criticality PIF"

As shown in Figure 2, the parameter criticality considers both high and low deviations from the nominal safe state. For example, a low level of reactor coolant system subcooling margin might indicate inadequate core cooling and an increased potential for core damage, while an excessive amount of subcooling might indicate an overcooling event and a potential pressurized thermal shock condition. The overall criticality of system condition PIF value is based on a weighted sum of the individual parameter criticality values Equation (3).

$$PIF_{System\ Criticality} = \frac{\sum_i \omega_i \ PIF_{Parameter\ Criticality\ _i}}{\sum_i \omega_i}$$

(Equation 3)

The ω_i value in Equation (3) is the weighting factor for parameter *i*. A higher value of the system criticality PIF indicates a more adverse overall plant condition. Additionally, the rate of change of the PIF value provides an indication if the overall plant health is improving or worsening.

2. Input File Format

```
SPDS_Input  number_of_parameters    Time_Lapse  time_lapse
SPDS_parameter_1  parameter_name_1
lo_lo_level_1     lo_level_1  hi_level_1  hi_hi_level_1
weighting_factor_1
.
.
SPDS_parameter_n  parameter_name_n
lo_lo_level_n     lo_level_n  hi_level_n  hi_hi_level_n
weighting_factor_n
```

3. Input File Description

number_of_parameters: Format: Integer. Range: ≥ 0. Specifies the number of procedure state prerequisites for the associated mental belief. If ***number_of_parameters*** is set to 0, no further safety parameter data should be entered. If ***number_of_parameters*** is greater than or equal to 1, input data for each safety parameter must be supplied or an error will be generated during input file processing.

time_lapse: Format: Double. Range: ≥ 0.0. Specifies the time interval between safety parameter updates. At each safety parameter update, the PIF value for each parameter is updated based on the operator's perceived information. If the operator has not perceived an updated value of the associated parameter since the last safety parameter update, the updated PIF value will be equal to the last value. For this reason, the time_lapse for safety parameter updates should not be set to a value less than the scanned parameter scan period (see "KB_OAT_Scanned_Parameter.txt").

SPDS_parameter: Format: String. Range: any. Specifies a unique identifying name for the associated safety parameter. It is recommended that the SPDS_parameter be set to the associated indicator name with the prefix "SPDS_" (i.e., SPDS_PZR_Level). Spaces must not be used - use the underbar character ("_") rather than a space when needed.

parameter_name: Format: String. Range: The parameter name must match a control panel parameter listed in the "ControlPanel.txt" input file. Specifies the control panel indicator that will be used to calculate the individual safety parameter PIF value. Spaces must not be used - use the underbar character ("_") rather than a space when needed.

lo_lo_level: Format: Double. Range: consistent with indicator output. Specifies the lower low level limit for the safety parameter. If the parameter value is less than the ***lo_lo_level***, the associated parameter PIF value is set equal to 10 (the maximum PIF value).

lo_level: Format: Double. Range: consistent with indicator output. Specifies the upper low level limit for the safety parameter. If the parameter value is between the ***lo_level*** and the ***hi_level***, the associated PIF value is set to 0. If the parameter value is between the ***lo_level*** and ***lo_lo_level***, the PIF value is calculated from linear interpolation between the low level limits (see Figure 2).

hi_level: Format: Double. Range: consistent with indicator output. Specifies the lower high level limit for the safety parameter. If the parameter value is between the ***lo_level*** and the ***hi_level***, the associated PIF value is set to 0. If the parameter value is between the ***hi_level*** and ***hi_hi_level***, the PIF value is calculated from linear interpolation between the high level limits (see Figure 2).

hi_hi_level: Format: Double. Range: consistent with indicator output. Specifies the upper high level limit for the safety parameter. If the parameter value is greater than the ***hi_hi_level***, the associated parameter PIF value is set equal to 10 (the maximum PIF value).

weighting_factor: Format: Double. Range: any (0.0 – 1.0 recommended). Specifies the individual weight for the associated safety parameter. The overall criticality of system condition PIF is calculated from the normalized weighted sum of all the individual safety parameter contributions (see Equation 3). Because the normalization process means that only the relative ratio of the safety parameter weighting factors to each other is important, the analyst can use any meaningful weighting factor scaling. Although this normalization process allows greater flexibility in assigning the weighting_factor, it is recommended that the ***weighting_factor*** be set between 0.0 and 1.0 to support easier interpretation of the file input

4. <u>Sample Input</u>

```
SPDS_Input  6     Time_Lapse        1.0
SPDS_Min_Sub_Cooling    Min_Sub_Cooling
5.0   25.0  75.0  100.0
1.0
SPDS_SG_A_WR_Level      SG_A_WR_Level
0.10  0.60  0.85  0.99
0.5
SPDS_PZR_Level          PZR_Level
0.10  0.40  0.70  0.95
0.75
```

In the above example, three parameters are used to calculate the criticality of system condition PIF: (1) minimum reactor coolant system subcooling margin (Min_Sub_Cooling), (2) steam generator A wide range level (SG_A_WR_Level), and (30 pressurizer level (PZR_Level). Based on the inputs, the nominal range for minimum subcooling is specified as 25.0 – 75.0 degrees F. Similarly, the nominal range for SG A wide range level is 0.60 – 0.85 and the nominal range for the pressurizer level is 0.4 - 0.7. Each safety parameter has been assigned a different weighting factor, with the subcooling margin identified as the dominant factor. SG A wide range level has half the weight of subcooling margin and the weight of the pressurizer level is 25% less than subcooling margin. The analyst can model an operator safety parameter priorities and unique knowledge base by adjusting these weighting factors.

Assuming a subcooling margin of 10 F, a SG A wide range level of 0.05, and a pressurizer level of 0.55, the system criticality PIF value would be calculated as follows:

$PIF_{subcooling}$ = 10.0 * (25.0 – 10.0)/(25.0 – 5.0) = 7.5
$PIF_{SG\ A\ WR\ Level}$ = 10.0
$PIF_{PZR\ Level}$ = 0.0

$PIF_{System\ Criticality}$ = (1.0 * 7.5) + (0.5 * 10.0) + (0.75 * 0.0)/(1.0 + 0.5 + .75)
= **5.56**

1. Purpose

In order for the operator to use information and plant data to support decision-making activities, raw information obtained from the control panel must pass through the operator's perception filters. Two perception filters are currently modeled in ADS-IDAC: (1) a quantitative filter which limits and focuses the ability of the operator to scan information on the control panel, and (2) a biasing filter which can distort parameter readings obtained by the operator. The biasing filter is described in the "KB_OAT_Bias_Factors.txt" section. The quantitative filter models the information scanning process used by operators to monitor plant status. In general, the control room operators frequently monitor a small subset of plant instrumentation. Typical parameters might include reactor power, reactor coolant system pressure and temperature, and critical inventory levels. When plant conditions degrade, the operators may add additional parameters to their scanning in response to procedural requirements or alarms. The more parameters the operator is able to monitor, they should be able to make a more accurate assessment of plant status. However, the operator does not have an infinite capacity to monitor plant instrumentation – consequently, there are limits to how many items and operator can effectively monitor. Additionally, external and internal factors (e.g., stress or time pressure) may force the operator to reduce the number of parameters monitored. Consequently, operators may limit monitoring activities to a focused set of items that are considered to be most pertinent to the perceived plant state.

Within the ADS-IDAC model, the focusing process is controlled by the operator's control panel "scan queue". The scan queue contains a listing of parameters that the operator monitors on a frequent basis. Scan queue parameters include instruments, alarms, and component states. The number of items contained in the scan queue is limited by the individual capabilities of the operator, the amount of attention the operator can apply to information gathering, and the operator's perception of the current plant state. As the number of monitored items in the scan queue increases, the operator improves their ability to accurately assess and diagnosis the plant state.

Two main factors determine which items are included in the operator's scan queue: (1) the maximum size limit of the queue, and (2) the priority level of each item in the queue. The maximum size of the scan queue ($N_{Scan\ Queue}$) is determined by Equation (4).

$$N_{Scan\,Queue} = N_{Baseline}\left(1-\gamma_1\,PIF_{Info\,Load}\right)\left(1-\gamma_2\,PIF_{System\,Criticality}\right) \quad \text{(Equation 4)}$$

The constants $N_{Baseline}$, γ_1, and γ_2 are set in each operator's profile and serve to calibrate the model to the desired operator performance level. $N_{Baseline}$ establishes the maximum amount of information that can be contained in the scan queue while the γ factors ($0 < \gamma_i < 0.1$) set the sensitivity of the dynamic scan queue limit to the information load and system criticality PIFs.

If the size of the scan queue exceeds $N_{Scan\ Queue}$, items are removed from the scan queue list until the size limitation is met. Two main factors determine which items will be removed from the scan list: (1) the items relevance to the operator's perceived plant state, and (2) the priority level of the item. Each indicator, control, and alarm on the control panel is associated with a set of high level plant functions in the input file "KB_OAT_System_Decomposition.txt". These plant functions directly correspond to a unique functional imbalance described in the file "KB_OAT_Event_Matrix.txt". When the operator determines that a functional imbalance exists, any items that are associated with the imbalance (as described in the system decomposition file) are considered to be relevant to the perceived plant state. When an item is assigned to the scan queue list, it is assigned an initial priority level (the priority level is equal to greater than 1, with 1 denoting the highest priority). As time progresses, an items priority level will decay if the item is not considered to be relevant to the perceived plant state. Thus, the priority level of relevant items will remain high, while the priority of non-relevant items will be gradually reduced. If an item's priority level is reduced below the priority limit, it will be removed from the scan queue. Furthermore, if the scan queue size exceeds the $N_{Scan\ Queue}$ limit, low priority items will be removed until the list size is within the limit.

The purpose of the "KB_OAT_Scanned_Parameter.txt" input file is to specify the parameters necessary for management of the operator scan queue. Additionally, the analyst can designate an initial list of parameters, components, and alarms that are monitored by the operator at the beginning of the simulation. This provides the means to model routine control panel scanning activities normally performed by the control room crew.

2. Input File Format

```
Scan_Period_(seconds)                 scan_period
Info_Load_Sensitivity                 information_load_sensitivty
System_Criticality_Sensitivty
      system_criticality_sensitivity
Priority_Decay_Time_(seconds)         priority_decay_time
Priority_Limit                        priority_limit
Relevance_Limit                       relevance_limit

Number_of_Scanned_Parameters          number_of_parameters
parameter_name_1                      parameter_priority_1
.                                     .
parameter_name_n                      parameter_priority_n

Number_of_Scanned_Components          number_of_components
component_name_1                      component_priority_1
.                                     .
component_name_n                      component_priority_n
```

```
Number_of_Scanned_Alarms          number_of_alarms
alarm_name_1                      alarm_priority_1
.                                 .
alarm_name_n                      alarm_priority_n
```

3. Input Description

scan_period: Format: Double. Range: ≥ 0.0. This parameter the specifies the time delay between control panel monitoring scans. A lower value for ***scan_period*** will improve the operator's plant state assessment making parameter updates more frequent. However, more frequent control panel scanning increases the operator's information load and may limit the number of parameters that can be monitored. Longer scan periods will minimize the information load but may result in the operator relying on outdated information, particularly during rapidly evolving plant events.

information_load_sensitivity: Format: Double. Range: $0.0 \leq x \leq 0.01$. This parameter specifies the sensitivity of the scan queue size limit to the operator's information load (the γ_1 factor in Equation 4). Because the maximum information load PIF value is 10.0, this factor is limited to values no greater than 0.01. The maximum size limit of the scan queue is set by parameter ***scan_queue_limit*** in the "ActionTaker.txt" input file.

system_criticality_senstivity: Format: Double. Range: $0.0 \leq x \leq 0.01$. This parameter specifies the sensitivity of the scan queue size limit to the operator's perceived criticality of the plant state (the γ_2 factor in Equation 4). Because the maximum criticality of system condition PIF value is 10.0, this factor is limited to values no greater than 0.01. The maximum size limit of the scan queue is set by parameter ***scan_queue_limit*** in the "ActionTaker.txt" input file.

priority_decay_time: Format: Double. Range: ≥ 0.0. Specifies the length of the delay interval (in seconds) for successive reductions in scanned item priority. The priority level of items in the scan queue will decrease over time (if the item is not considered to be relevant to the operator). When a scanned item's priority level exceeds the ***priority_limit***, the item will be removed from the scan queue list. Therefore, the ***priority_decay_time*** effectively controls how long a scanned item resides in the scan queue list.

priority_limit: Format: Integer. Range: ≥ 0. This factor specifies the lowest item priority level that can be maintained in the operator's scan queue. In ADS-IDAC, high priority items are designated by lower priority values (i.e., the highest priority items have a priority level of 1). As the priority of an item decreases, its priority level increases. If the priority level of an item is greater than the ***priority_limit***, it will be removed from the operator's scan queue.

relevance_limit: Format: Double. Range: $0.0 \leq x \leq 1.0$. This parameter specifies the threshold value used by the operator model to determine if an imbalance event diagnosis is relevant. Each item listed in the scan queue is linked to one or more associated imbalance events in the "KB_OAT_System_Decomposition.txt" input file. If the maximum imbalance event diagnostic score for the functions supported by the item is greater than the relevance_limit, the scanned item will be considered to be relevant to the operator. If the scanned item is not relevant to the operator, the item's priority level will be decreased one increment (i.e., the item's priority will be increased by 1) during every ***priority_decay_time*** interval.

number_of_parameters: Format: Integer. Range: ≥ 0. Specifies the number of parameters that will be included in the operator scan queue at the start of the ADS-IDAC simulation. If ***number_of_parameters*** is set to 0, no further parameter data should be entered. If ***number_of_parameters*** is greater than or equal to 1, input data for each scanned parameter must be supplied or an error will be generated during input file processing. Because operators cannot passively gather parameter value information (i.e., control panel indicators must be actively read by the operator), parameters that would normally be known by the operator at the start of the simulation should be included in the parameter scan queue list.

parameter_name: Format: String. Range: The ***parameter_name*** must match a control panel parameter listed in the "ControlPanel.txt" input file. Contains the name of the parameter to be included in the scan queue. Spaces must not be used in the ***parameter_name*** (the underbar character "_" may be substituted for a space when needed).

parameter_priority: Format: Double. Range: ≥ 1.0. Specifies the parameter's initial priority level. A lower ***parameter_priority*** designates a higher priority parameter (i.e., the highest priority items are designated with a priority level of 1). Because the priority level can decay over time, specifying a low ***parameter_priority*** level will increase the residence time of the parameter on the scan queue list.

number_of_components: Format: Integer. Range: ≥ 0. Specifies the number of component states that will be included in the operator scan queue at the start of the ADS-IDAC simulation. If ***number_of_components*** is set to 0, no further component data should be entered. If ***number_of_components*** is greater than or equal to 1, input data for each scanned component state must be supplied or an error will be generated during input file processing. Because operators cannot passively gather component state information (i.e., component states must be actively read by the operator on the control panel), component states that would normally be known by the operator at the start of the simulation should be included in the parameter scan queue list.

component_name: Format: String. Range: any. Contains the name of the component to be included in the scan queue. The ***component_name*** must match a control panel component listed in the "ControlPanel.txt" input file. Spaces must not be used in the ***component_name*** (the underbar character "_" may be substituted for a space when needed).

component_priority: Format: Double. Range: $\geq$ 1.0. Specifies the component's initial priority level. A lower ***component_priority*** designates a higher priority component (i.e., the highest priority items are designated with a priority level of 1). Because the priority level can decay over time, specifying a lower ***component_priority*** will increase the residence time of the component on the scan queue list.

number_of_alarms: Format: Integer. Range: $\geq$ 0. Specifies the number of alarms that will be included in the operator scan queue at the start of the ADS-IDAC simulation. If ***number_of_alarms*** is set to 0, no further safety parameter data should be entered. If ***number_of_alarms*** is greater than or equal to 1, input data for each scanned alarm must be supplied or an error will be generated during input file processing. Although an alarm actuation is one method for an operator to passively gather information, an alarm state cannot be perceived until either the alarm actuates or the operator actively verifies the alarm state. For example, the activation of a mental belief that includes a prerequisite condition that an alarm is <u>not</u> actuated will require that (1) the alarm had previously actuated and cleared, or (2) the operator actively interrogated the alarm state. Therefore, it is recommended that any alarms listed in the "KB_OAT_HardWiredDiagnosis.txt" with an expected state of "OFF" (i.e., integer code 3023, "VOFF") be included in the initial alarm scan list.

alarm_name: Format: String. Range: any. Contains the name of the parameter to be biased. The parameter name must match a control panel parameter listed in the "ControlPanel.txt" input file. Spaces must not be used in the ***alarm_name*** (the underbar character "_" may be substituted for a space when needed).

alarm_priority: Format: Double. Range: $\geq$ 1.0. Specifies the alarm's initial priority level. A lower ***alarm_priority*** level designates a higher priority alarm (i.e., the highest priority items are designated with a priority level of 1). Because the priority level can decay over time, specifying a lower ***alarm_priority*** will increase the residence time of the alarm on the scan queue list.

4. <u>Sample Input</u>

```
Scan_Period_(seconds)              30.0
Info_Load_Sensitivity              0.05
System_Criticality_Sensitivty      0.05
Priority_Decay_Time_(seconds)      300.0
Relevance_Limit                    0.3

Number_of_Scanned_Parameters  6
```

```
SUR                     3
Core_Power              1
RATE_Core_Power         2
Loop_A_Tave             4
Loop_B_Tave             4
Loop_C_Tave             4

Number_of_Scanned_Components  3
Reactor_Trip            1
Safety_Injection        1
Turbine_Trip            1

Number_of_Scanned_Alarms      2
A_SG_A_Level_Lo_Dev     3
A_SG_A_Level_Hi_Dev     3
```

This input file places eleven items on the operator's initial scan queue list (six parameter values, three component states, and two alarm states). The operator will update the perceived value of these items every 30 seconds (***scan_period*** = 30.0). A sensitivity factor of 0.05 is applied to both the information load and system criticality PIFs (the maximum allowable size of the scan queue is set by the ***scan_queue_limit*** parameter in the "ActionTaker.txt" input file). Assuming the maximum allowable scan queue size of 30 items, if the information load PIF value was 5.0 and the system criticality PIF was 10, these sensitivity factors would limit the scan queue size to eleven items:

$$
\begin{aligned}
N_{scan\ queue} &= 30 * (1 - 0.05 * 5.0) * (1 - 0.05 * 10.0) \\
&= 30 * (1 - 0.25) * (1 - 0.5) \\
&= 30 * (0.75) * (0.5) = 11.25
\end{aligned}
$$

Every 300 seconds (i.e., the ***priority_decay_time***), the priority level items in the scan queue would be reduced by one increment if their associated relevance factor was less than the 0.3 ***relevance_limit***.

1. Purpose

In order to model an operator's mental model of the reactor plant, a functional decomposition of reactor plant systems and components is used to link parameters, components, alarms, and controls that support similar functions. The "KB_OAT_System_Decomposition.txt" input file identifies the key reactor plant functions and maps each control panel indicator, control, and alarm to these functions. The system decomposition supports two main features in the ADS-IDAC model: (1) skipping of procedure steps, and (2) management of the operator scan queue. These features are supported by the assignment of a "relevance factor" (in the range of 0.0 – 1.0) to each component based on the operator's perceived assessment of the plant state. Items that are considered to be relevant to the current plant state assessment are assigned a high relevance factor, while items considered to be not relevant are assigned a low value. The relevance is based on the diagnosis score for the functional imbalance(s) associated with the component. Thus if the operator has diagnosed that a specific imbalance event has occurred, items associated with the functional imbalance will be assigned high relevance score. For procedure step skipping, actions associated with items relevant to the plant state assessment will be less likely to be skipped, while actions associated with non-relevant items may be more likely to be skipped. Similarly, the priority level of items in the scan queue list that are relevant will remain at a higher level and are more likely to be retained in the scan queue.

ADS-IDAC utilizes a functional component categorization based on the flow of energy, mass, and momentum. In this modeling scheme, the reactor plant is viewed as a collection of mass, energy, and momentum flow paths, each containing sources and sinks. For example, in a PWR, the reactor core is considered to be a source of energy, while each steam generator is considered to be an energy sink. Because the reactor coolant system carries the energy released in the reactor core to the steam generators, any imbalance between energy production and removal will impact the reactor coolant energy state. In general, the following rules are used to identify mass, energy, and momentum imbalances:

- Energy flow imbalances are generally indicated by changes in temperature for subcooled single phase systems and changes in pressure for saturated two phase systems;
- Imbalances between mass sources and sinks are generally related to net inventory measures such as tank or vessel levels; and
- Momentum imbalances are generally indicated by changes in flow rates.

This modeling technique provides a powerful mechanism for linking components within a functional framework. In order to functionally categorize plant components, it is first necessary to identify the flow path boundaries. Plant system groups are used to represent the boundaries for mass, energy, and momentum flow paths. In general, it is desirable to make the plant system group boundaries as broad as possible in order to maximize the ability to link plant components within the operator knowledge base.

The strong coupling among nuclear plant systems presents a significant challenge when identifying functional system groups. Within a nuclear plant, energy flow is often carried by moving fluids such as the reactor coolant or main steam systems; there-fore, changes in mass flow rate can directly impact energy flow. Consequently, coupling can result in imbalances in one flow type influencing a second flow type within the same system group or a connected system group. Coupling can also mask the cause of disruption in energy, mass, or momentum flow. For example, changes in reactor coolant system temperature due to an imbalance between reactor core power and turbine load (an energy flow imbalance) can result in variations in system volume due to the expansion or contraction of the coolant (which might be interpreted as a mass flow imbalance). An additional consideration is the diagnostic capability afforded by the system groupings. It is desirable to constrain the system group boundaries such that a flow imbalance within a grouping can be linked to a manageable number of potential causes. In practice, the identification of the system groups requires a balance between maximizing the linkage between plant components, minimizing undesirable coupling, and providing a high level of diagnosticity. For a pressurized water reactor model, it is recommended that five functional system groups be used (Table 5).

Table 5. Pressurized Water Reactor
System Groups and Flow Paths

System Group	Flow Paths
Reactor Coolant	Energy
	Mass
	Momentum
Pressurizer	Energy
	Mass
Steam Generators[(1)]	Energy
	Mass
Secondary[(2)]	Energy
	Mass
Containment	Energy

(1) Each steam generator is considered a separate system group
(2) The secondary system group includes the turbine, main steam, main feed, and condenser systems.

The component map describes the functions associated with every control, indicator, and alarm available to the ADS-IDAC control room crew. Each operator knowledge base includes a unique component functional map in order to match operator behavior with to a desired level of knowledge, skills, and abilities. A three parameter coding scheme is used to identify component functions. The first parameter identifies the type of flow (i.e., energy, mass, or momentum). The second parameter identifies the system group that transports the energy, mass, or momentum flow. The third parameter identifies how the component affects (or is associated with) the flow balance in the system group. Thus, a possible component functional code might read: "energy flow, reactor coolant system, energy source". More than one functional code can be as-signed for a single component. The "KB_OAT_System_Decomposition.txt" component functional map allows each component to be meaningfully linked within the operator knowledge base.

2. Input File Format

```
"text_desription_of_functional_decomposition"

Number_of_Functional_Items    number_of_functions
function_name_1               function_code_1
.                             .
function_name_n               function_code_n

Control_Panel_Decomposition
item_name_1       item_function_1 ....    item_function_j    999
.                 .                       .
item_name_m       item_function_1 ....    item_function_j    999
```

3. Input Description

text_desription_of_functional_decomposition: Format: String. Range: any. The analyst may enter any desired text into this field. However, the entire field entry must be enclosed in quotation marks (quotation marks may not be used within the field entry) ADS_IDAC recognizes quotation marks as a special character delineating the functional decomposition description. This field provides the analyst to describe the reactor plant functional decomposition in plain language to improve the clarity of input data. Because ADS-IDAC utilizes special integer codes to identify component functions (rather than text strings), this field can be used to provide a plain language "roadmap" between the function codes and function description. To improve input file clarity and reduce data entry errors, it is recommended that the functional codes be assigned using a consistent framework. One such framework uses three digit integers where each digit position has a defined purpose (as described in the Sample Input section). Although a text entry is optional, the analyst should supply a set of empty quotes (" ") if no text description will be provided to avoid an error during input file processing.

number_of_functions: Format: Integer. Range: ≥ 0. Specifies the number of functions that will be used by the operator knowledge base. If ***number_of_functions*** is set to 0, no further function data should be entered. If ***number_of_functions*** is greater than or equal to 1, input data for each function (i.e., ***function_name*** and ***function_code***) must be supplied or an error will be generated during input file processing.

function_name: Format: String. Range: any. This parameter is used to specify the unique description name for each function. Each ***function_name*** should be associated with a imbalance event included in the "KB_OAT_Event_Matrix.txt" input file. Although the simulation will not stop if ADS-IDAC is unable to locate the ***function_name*** in the "KB_OAT_Event_Matrix.txt", an error message will be generated and the relevance factor will be set to 0.0. ADS-IDAC recognizes the special function_name "Not_Applicable" – this function should be used for items meant to provide executive control over the simulation (such as timers or other items that the analyst wishes to always be assigned a relevance factor of 1.0).

function_code: Format: Integer. Range: any. Specifies a unique identifying code for each ***function_name.*** The ***function_code*** for each entry should be unique (codes should not be associated with more than one ***function_name***). It is recommended that the functional codes be assigned using a consistent framework such as using a multidigit integers where each digit position has a defined purpose (e.g., first digit refers to function, second digit refers to plant system, third digit refers to imbalance trend, etc.). Only one ***function_code*** is assigned to each ***function_name***.

item_name: Format: String. Range: any. Specifies the specific control panel control, indicator, component, or alarm that will be described by the date entry. Each item included in the "ControlPanel.txt" file must have a functional description in both the ActionTaker and Decision Maker knowledge bases or an error will be generated during input file processing.

item_function: Format: Integer. Range: must match a ***function_code***. More than one code may be entered for a component. It is not necessary to enter a function_code, but if no codes are specified, the associated control panel item will always have a relevance score of 0.0. If an item is assigned the integer code for the "Not_Applicable" function (a special text code recognized by ADS-IDAC), the relevance factor for the item will be set equal to 1.0.

Note: Each item data entry line should be terminated with the integer code "999". This code is recognized by ADS-IDAC as the termination of the item functional description. A "999" termination entry should be used even if no ***item_function*** codes are entered for the item.

4. <u>Sample Input</u>

```
"Functional Decomposition:
   Function Type (First Digit)
      1 - Mass Imbalance
      2 - Energy Imbalance
      3 - Momentum Imbalance
   System Type (Second Digit)
```

```
      1 – Reactor Coolant System
      2 – Pressurizer
      3 – Steam Generator A
      4 – Steam Generator B
      5 – Steam Generator C
      6 – Secondary System (Turbine, Condenser)
      8 – Containment
   Imbalance Trend (Third Digit)
      1 – Decrease
      2 – Increase
   Special Codes
     900 – Not Applicable"
Number_of_Functional_Items    11
Mass_Imbalance_RCS_Low              111
Mass_Imbalance_PZR_Low              121
Mass_Imbalance_SG_A_Low             131
Mass_Imbalance_SG_B_Low             141
Mass_Imbalance_SG_C_Low             151
Energy_Imbalance_RCS_Low            211
Energy_Imbalance_PZR_Low            221
Energy_Imbalance_SG_A_Low           231
Energy_Imbalance_SG_B_Low           241
Energy_Imbalance_SG_C_Low           251
Not_Applicable                      900

Control_Panel_Decomposition
Time                    900   999
Watchdog_Timer          900   999
PZR_Level               111   121   999
PZR_Level_Setpoint      111   121   999
ECCS_Flow               111   121   211   999
```

This sample input utilizes a set of three digit functional codes to describe the system functional decomposition. The decomposition structure is described in the ***text_desription_of_functional_decomposition*** field. The decomposition consists of three main function types (mass balance, energy balance, and momentum balance), eight primary system groups, and two imbalance trends. The special function code "Not_Applicable" is also used as the functional code for certain items used to provide executive control over the simulation (i.e., Time and Watchdog_Timer). The analyst may assign any functional integer code to the function "Not_Applicable". The three digit function code is interrupted as follows: the first digit describes the function type, the second digit provides the associated system group, and the third digit provides the trend direction. Thus the integer code "241" is interrupted as an decreasing trend in SG B energy. The input file provides specific functional codes for eleven imbalance events – each of these events should have a companion entry in the "KB_OAT_Event_Matrix.txt" input file. In general, the analyst should designate functional codes for every function that can be described by the selected decomposition framework. In this example, three function, eight system groups, and two trends would require 3 x 8 x 2 = 48 separate functional entries.

The sample input also provides several examples of control panel item functional descriptions. In this example, the parameter "ECCS_Flow" has been associated with the functions 111 (low mass in reactor coolant system), 121 (low mass in the pressurizer), and 211 (low energy in the reactor coolant system). Thus, if the operator has diagnosed a low mass condition in the pressurizer (based on the symptoms described in the "KB_OAT_Event_Matrix.txt" input file), the parameter "ECCS_Flow" will be assigned a high relevance factor. Consequently, actions associated with monitoring ECCS_Flow will be less likely to be skipped and the ECCS_Flow parameter, if it is listed in the scan queue, will be more likely to be retained in the monitoring list.

1. Purpose

Each operator profile includes data to define how the time constraint load PIF value is calculated. The profile contains a listing of plant parameters used to calculate the time constraint PIF value along with the associated critical threshold values. Typical parameters to calculate this PIF include steam generator water levels, pressurizer water level, and reactor coolant system pressure. Two different threshold levels are used to calculate the PIF value – a normal operation threshold and an accident threshold. When the operator's high level goal is maintaining normal operation or troubleshooting an abnormal condition, the normal operation threshold is used. If the operator switches to the goal of mitigating an accident condition, the time constraint PIF value is based on the accident threshold. The use of two different threshold values allows ADS-IDAC to capture an operator's changing sensitivity to key parameters depending on the overall perceived plant condition. In general, the normal accident threshold is set to a level corresponding to reactor plant trip set points. The accident level threshold is normally set to a less restrictive value that is more indicative of the availability of a key safety function. For example, if a plant that has an automatic reactor trip on low steam generator (SG) water level, an operator might focus on the time available until the reactor trip set point is reached during an uncontrolled decrease in SG level. However, once the reactor is tripped, the operator's focus might shift to simply maintaining adequate decay heat removal capability from the steam generator - a function that can be often be performed with a much lower SG level. Thus the normal threshold might be set equal to the low SG water level reactor trip set point while a less restrictive value is used for the accident threshold.

The time constraint load PIF value is based on information perceived by the operator rather than data obtained directly from the thermal-hydraulic model. Perceived data will differ from the actual parameter value in thermal-hydraulic model due to time lags in updating perceived data and any distortion introduced by the perception filter. The first step in determining the time constraint loading PIF is to determine the time available until each time constrained parameter exceeds a critical threshold Equation (5).

$$t_{i,available} = \frac{P_i - P_{i,Threshold}}{\dot{P}_i} \qquad \text{(Equation 5)}$$

In Equation 2a, $t_{i,available}$ is the time until the value of parameter *i* (P_i) exceeds threshold value $P_{i,Threshold}$. The minimum value of $t_{i,available}$ (denoted as $t_{min,i}$) is then used to calculate the time constraint load PIF for the parameter Equation (6).

$$PIF_{Time\,Constraint,i} = 10\left[1 - \frac{(t_{\min,i} - t_{Lower})}{(t_{Upper} - t_{Lower})}\right] \qquad \text{(Equation 6)}$$

The tuning constants t_{Lower} and t_{Upper} are used to calibrate the PIF value to the desired operator characteristics. Similar to the information loading PIF, if the minimum time available exceeds t_{Upper}, the time constraint PIF value is set to 0. If the minimum time available is less than t_{Lower}, the PIF value is assumed to saturate at a value of 10.

In order to more realistically model dynamic changes in the time constrained PIF factor, the updated PIF value is passed through a lag filter to simulate the buildup and decay of stress associated with time constrained loading. If the parameter value has not crossed the lower accident threshold value, the filter output is determined from the following equation (7):

$$PIF_{TCL}^{Updated} = PIF_{TCL}^{Old} + (PIF_{TCL}^{New} - PIF_{TCL}^{Old})\left(\frac{t_{lapse}}{\tau_{buildup}}\right) \qquad \text{(Equation 7)}$$

Where

PIF^{Old} is last updated PIF value,
$PIF^{Updated}$ is the output from the lag filter
PIF^{New} is the unfiltered PIF value
t_{lapse} is the update periodicity (in seconds), and
$\tau_{buildup}$ is the buildup time constant (in seconds).

Once the parameter passes the accident threshold, the parameter PIF value is allowed to decay using the following formula (Equation 8):

$$PIF_{TCL}^{Updated} = 10(1 - e^{(t_{current} - t_{threshold})/\tau_{decay}}) \qquad \text{(Equation 8)}$$

Where

$PIF^{Updated}$ is the output from the lag filter
$t_{current}$ is the current simulation time
$t_{threshold}$ is the time that the parameter exceeded the accident threshold value
τ_{decay} is the decay time constant

The overall time constraint PIF factor for the operator is set equal to the maximum parameter time constrained PIF value (i.e., the most limiting parameter establishes the PIF value for the operator).

2. Input File Format

```
Num_of_Time_Constrained_Parameters   number_of_parameters
Time_Lapse_For_Update                time_lapse
Lower_Threshold_(minutes)            lower_threshold
```

Input File: “KB_OAT_Time_Constrained_Input.txt”

```
Upper_Threshold_(minutes)           upper_threshold
Decay_Time_Constant_(sec)           decay_time_constant
Buildup_Time_Constant_(time_steps)  buildup_time_constant

time_constrained_parameter_1                 parameter_name_1
nominal_value_1   normal_threshold_1         accident_threshold_1
.
.
time_constrained_parameter_n                 parameter_name_n
nominal_value_n   normal_threshold_n         accident_threshold_n
```

3. Input Description

number_of_parameters: Format: Integer. Range: ≥ 0. This parameter specifies the number of time constraint load parameters included in the "KB_OAT_Time_Constrained_Input.txt" file. The ***number_of_parameters*** must equal the actual number of time constrained parameters listed in this file.

time_lapse: Format: Double. Range: ≥ 0. Specifies the time increment (in seconds) between successive updates in time constrained parameter values. Because time constrained parameters are based on perceived information, the update uses the operator's currently memorized value. If the operator has not perceived a new parameter value since the last update, an updated value is calculated from the perceived rate of change of the parameter value.

lower_threshold: Format: Double. Range: ≥ 0. Specifies the lower threshold value used to calculate the time constraint PIF factor for each monitored parameter. If the time to reach the applicable threshold value is less than the ***lower_threshold,*** the parameter PIF value is set to 10.0. If the time to reach between the ***lower_threshold*** and ***upper_threshold***, the PIF value is interpolated based on the parameter value and rate of change.

upper_threshold: Format: Double. Range: ≥ 0. Specifies the upper threshold value used to calculate the time constraint PIF factor for each monitored parameter. If the time to reach the applicable threshold value is greater than the ***upper_threshold,*** the parameter PIF value is set to 0.0. If the time to reach between the ***lower_threshold*** and ***upper_threshold***, the PIF value is interpolated based on the parameter value and rate of change.

decay_time_constant: Format: Double. Range: ≥ 0. Specifies the decay constant to be applied to the parameter PIF value when the associated parameter exceeds the accident threshold.

buildup_time_constant Format: Double. Range: ≥ 0. Specifies the buildup time constant to be applied to the parameter PIF value. This parameter allows a more realistic buildup and decay of time constrained induced stress.

time_constrained_parameter: Format: String. Range: any. Descriptive name for the time constrained parameter. Spaces must not be used (use the underbar character ("_") rather than a space when needed. It is recommended that time constrained parameters be given the prefix "TCL_" to improve the readability of the output files.

parameter_name: Format: String. Range: Must match a ***parameter_name*** listed in the "ControlPanel.txt" input file. Specifies the monitored parameter for the time constrained factor.

nominal_value: Format: Double. Range: any, should be consistent with the monitored parameter. Specifies the nominal value for the monitored parameter. This parameter is used to determine if the normal parameter value is above or below the applicable threshold values.

normal_threshold: Format: Double. Range: any, should be consistent with the monitored parameter. Establishes the parameter threshold value used during normal operation. For the purposes of calculating the time constrained loading factor, normal operation is considered to apply for any crew goal except: (1) maintaining global safety margin or (2) troubleshooting when the reactor is tripped.

accident_threshold: Format: Double. Range: any, should be consistent with the monitored parameter. Establishes the parameter threshold value used during emergency operation. For the purposes of calculating the time constrained loading factor, emergency operation is considered to apply when the crew goal is: (1) maintaining global safety margin or (2) troubleshooting when the reactor is tripped.

4. Sample Input

```
Num_of_Time_Constrained_Parameters          3
Time_Lapse_For_Update                       30.0
Lower_Threshold_(minutes)                   5.0
Upper_Threshold_(minutes)                   25.0
Decay_Time_Constant_(sec)                   100.0
Buildup_Time_Constant_(time_steps)          50.0
TCL_PZR_Pressure         PZR_Pressure
2250.      2000.         750.
TCL_PZR_Level            PZR_Level
0.50       .20           .10
TCL_SG_A_NR_Level        SG_A_NR_Level
0.44       .25           .10
```

This sample defines three parameters that are used to calculate the overall time constraint load factor:

- pressurizer pressure (PZR_Pressure): The nominal value is 2250 psig, the normal operating threshold value is 2000 psig (a value just above the reactor trip setpoint), and the accident threshold is 750 psig (a value just above the pressure at which the safety injection accumulators passively inject cooling water).

- pressurizer level (PZR_Level): The nominal value is 0.50 (50% water level), the normal operating threshold is 20% (a value just below the lower range of the controller range), and the accident threshold is 10% (the crew should initiate safety injection at a level of 0.12).

- steam generator A narrow range level (SG_A_NR_Level): The nominal level is 44%, the normal operating threshold is 25% (a level just below the lower end of the normal controller range), and the accident threshold is 10% (the minimum level required for the SG to be credited as a heat sink)

Parameter values are updated every 30 seconds (***time_lapse_for_update***). If the time to reach a threshold value is less then 5 minutes, the PIF factor is set to 10.0 (***lower_threshold***). If the time to reach a threshold is greater than 25 minutes, the PIF factor is set to 0 (***upper_threshold***).

Operator Name: Decision Maker (ODM)

Responsibilities:

The Decision Maker fulfills the role of control room supervisor (or senior reactor operator) within the nuclear plant control room environment. The Decision Maker is responsible for directing all written procedures (e.g., emergency operating procedures) and selecting high level crew goals. However, the Decision Maker does not directly manipulate control panel controls. Instead, the Decision Maker directs the Action Taker to execute control manipulations and provide indicator readings. Although the Decision Maker can implement mental procedures, because of the limitation that all plant interactions must be executed by the Action Taker, the Decision Maker cannot independently perform activities. The Decision Maker's selection of high level crew goals influencing the overall problem solving strategy (e.g., following written procedures or executing knowledge-based problem solving). Additionally, only the Decision Maker can initiate transitions between written procedures (e.g., between emergency operating procedures, or entry into functional recovery guidelines). These input files, in conjunction with the procedure step input files, constitute the knowledge base for the Decision Maker. The knowledge base is intended to mode the skills, abilities, and experience of the operator.

Input Files:

The following files have the same form and content as the Action Taker input file. There are two main differences between the Action Taker and the Decision Maker input files: (1) Decision Maker input files are located in the Input/Crew/DecisionMaker project file directory (rather than the Input/Crew/ActionTaker directory), and (2) in general, the Decision Maker file names replace the acronym "OAT" with "ODM" to designate that the files are associated with the Decision Maker. Additionally, the DecisionMaker.txt input file has the same form and content as the ActionTaker.txt input file. Refer to the Action Taker input file descriptions earlier in this section for a detailed description of the following input files.

- DecisionMaker.txt
- KB_ODM_Bias_Factors.txt
- KB_ODM_Event_Matrix.txt
- KB_ODM_HardWired_Diagnosis.txt
- KB_ODM_Safety_Parameters.txt
- KB_ODM_Scanned_Parameters.txt
- KB_ODM_System_Decomposition
- KB_ODM_Time_Constrained_Input.txt

Because only the Decision Maker can direct knowledge-based actions, a more complete file description of the "KB_ODM_Diagnosis_Actions.txt" input file is included in this section.

1. Purpose

Diagnosis actions are used to implement the knowledge-based problem solving approach when the control room crew is implementing the “troubleshooting” goal. This problem-solving approach is intended to model knowledge-based actions that an operator might perform outside the scope of the emergency procedures. Diagnostic actions are grouped within functional areas that are aligned with the imbalance events described in the “KB_OAT_Event_Matrix.txt” file. Because many possible actions may be available to address a specific imbalance event, each diagnosis action is assigned a priority level and a set of prerequisite conditions. Based on their perception of the plant state, operators might execute actions they believe to be reasonable given their situational assessment but are not necessarily covered by plant procedures. Examples of such actions include reducing reactor coolant system water injection when pressurizer level is high or decreasing the steam dump rate when steam generator pressure is low. Within ADS-IDAC, knowledge-based actions can be activated when, based on the operator’s perceived plant state, the event membership value of a functional imbalance diagnosis exceeds a pre-defined threshold value. For example, an imbalance diagnosis of “low mass in the reactor coolant system” might lead an operator to increase reactor coolant system injection flow, reduce normal letdown flow, or actuate emergency core cooling systems. Knowledge-based actions have the following characteristics and properties in the ADS-IDAC model:

- Action rules are organized within functional imbalance diagnostic groups. Each possible functional imbalance event can be associated with a list of actions intended to mitigate the associated mass, energy, or momentum imbalance.

- Each functional imbalance diagnosis group is as-signed a priority level in order to reflect the relative importance of the associated actions to the operator. For example, actions intended to address inadequate core cooling might be sequenced before actions to address low steam generator inventory in a single steam generator. The priority can be adjusted to reflect an opera-tor’s knowledge, experience, and problem solving style.

- Each action can be assigned a set of prerequisite conditions that must be met prior to execution of the action. Prerequisites are used to better model the heuristic rules an operator might use to activate a specific action. Prerequisites can be associated with plant parameters, component states, alarms, active procedures in use, or an operator’s mental beliefs.

- Once an action in a functional diagnosis group has been activated, further actions within the functional area will be blocked for a pre-defined dormancy period. The dormancy period allows the operator to address other, possibly lower priority, functional areas.

As an example of the implementation of knowledge-based actions, consider the depressurization of reactor coolant system following a steam generator tube rupture. The EOPs direct the operator to reduce primary pressure in order to equalize reactor coolant system pressure and ruptured steam generator pressure. This action reduces coolant leakage through the ruptured steam generator tube and facilitates re-fill of the reactor coolant system. Although a knowledge-based paradigm for execution of this action will not match the efficiency and stability afforded by the EOPs, knowledge based rules can be used to achieve a similar end state.

In the current version of ADS-IDAC, all knowledge-based actions are initiated by the Decision Maker; therefore the diagnosis action file for the Action Taker does not affect the simulation.

2. Input File Format

```
number_of_diagnoses       number_of_imbalance_diagnoses

diagnosis_name            imbalance_diagnosis_name_1
diagnosis_priority        diagnosis_priority
reset_delay_time          minimum_time                weibull_alpha
      weibull_beta
number_of_actions         number_of_actions

      action_name         action_name_1
      action_priority     action_priority_1
      action_type         action_type_1
      control_input       [control_input_value_1]
      [control_parameter_1]
      lower_limit         lower_control_limit_1
      upper_limit         upper_control_limit_1
      number_of_prerequisites number_of_prerequsites
            item_name_1       [parameter_1
                  parameter_2]
            [verification}
            [minimum_time     weibull_alpha
                  weibull_beta]
            expected_state    [relationship     value_1
                  value_2]
            logic_flag
            .                 .
            .                 .
            item_name_n       [parameter_1
                  parameter_2]
            [verification}
            [minimum_time     weibull_alpha
                  weibull_beta]
```

```
        expected_state    [relationship     value_1
              value_2]
        logic_flag

  .
  .
  .

diagnosis_name         imbalance_diagnosis_name_n
diagnosis_priority     diagnosis_priority
reset_delay_time       double_1    double_2    double_3
number_of_actions      number_of_actions

      action_name       action_name_1
      action_priority   action_priority_1
      action_type       action_type_1
      control_input     [control_input_value_1]
      [control_parameter_1]
      lower_limit       lower_control_limit_1
      upper_limit       upper_control_limit_1
      number_of_prerequisites number_of_prerequsites
            item_name_1      [parameter_1
                  parameter_2]
            [verification}
            [minimum_time     weibull_alpha
                  weibull_beta]
            expected_state    [relationship     value_1
                  value_2]
            logic_flag
            .                 .
            .                 .
            item_name_n      [parameter_1
                  parameter_2]
            [verification}
            [minimum_time     weibull_alpha
                  weibull_beta]
            expected_state    [relationship     value_1
                  value_2]
            logic_flag

      .
      .
      .
```

3. Input Description

number_of_diagnoses: Format: Integer. Range: ≥ 0. This parameter specifies the number of diagnosis event categories used for knowledge-based actions. The ***number_of_diagnoses*** must equal the actual number of diagnosis event categories entered in this input file.

imbalance_diagnosis_name: Format: String. Range: Must match a valid ***event_name*** of diagnosis ***type*** “Imbalance” in the “KB_ODM_Event_Matrix.txt” input file. Specifies the event diagnosis used to activate the associated knowledge-based actions.

diagnosis_priority: Format: Integer. Range: ≥ 1. Specifies the priority level of the associated ***imbalance_diagnosis_name***. A lower ***diagnosis_priority*** value designates a higher priority imbalance diagnosis (i.e., the highest priority items are designated with a priority level of 1).

minimum_time: Format: double. Range: > 0.0. In conjunction with the ***weibull_alpha*** and ***weibull_beta*** variables, specifies the: (1) dormancy time for the associated diagnosis actions following activation, and (2) the required to evaluate action prerequisites. In order to capture the uncertainty associated with the these times, a three parameter Weibull probability distribution is used. The Weibull distribution is given by the following equation (9):

$$F(t) = 1 - \exp\left[-\left(\frac{t-u}{\alpha} \right)^{\beta} \right]$$

(Equation 9)

Where:

minimum_time = u (minimum time), seconds
weibull_alpha = α parameter
weibull_beta = β parameter

In addition to specifying the reset delay time for the associated imbalance event, this parameter should be provided for all prerequisite data fields except when the prerequisite ***item_name*** is set to the reserved word **Mental_Belief**.

weibull_alpha: Format: double. Range: > 0.0. In conjunction with the ***weibull_alpha*** and ***weibull_beta*** variables, specifies the: (1) dormancy time for the associated diagnosis actions following activation, and (2) the required to evaluate action prerequisites. In addition to specifying the reset delay time for the associated imbalance event, this parameter should be provided for all prerequisite data fields except when the prerequisite item_name is set to the reserved word **Mental_Belief**.

weibull_beta: Format: double. Range: > 0.0. In conjunction with the ***weibull_alpha*** and ***weibull_beta*** variables, specifies the: (1) dormancy time for the associated diagnosis actions following activation, and (2) the required to evaluate action prerequisites. In addition to specifying the reset delay time for the associated imbalance event, this parameter should be provided for all prerequisite data fields except when the prerequisite item_name is set to the reserved word **Mental_Belief**.

number_of_actions: Format: Integer. Range: ≥ 0. This parameter specifies the number of knowledge-based actions associated with the

`imbalance_diagnosis_name`. The ***number_of_actions*** must equal the actual number of actions entered for the imbalance event.

`action_name:` Format: String. Range: Must match a valid controller listed in the "ControlPanel.txt" input file. Specifies the controller that is manipulated by the knowledge-based action.

`action_priority:` Format: Integer. Range: ≥ 1. Specifies the priority level of the associated ***`action_name`***. A lower ***action_priority*** value designates a higher priority imbalance diagnosis (i.e., the highest priority items are designated with a priority level of 1). Higher priority actions are performed prior to lower priority actions.

`action_type:` Format: Integer. Range: Should refer to one of the following valid integer codes:

Integer Code	Description	Comments
3074	VACTION	Simple controller manipulation. The ***action_name*** controller will be positioned to the value specified by ***control_input***
3112	VADDITIVE	The ***action_name*** controller control value will be incremented by the amount specified by the ***control_input*** variable.
3139	VPARAMETER_CONTROL	The ***action_name*** controller will be positioned to the perceived value of the ***control_parameter*** indicator. If the ***control_parameter*** value has not been perceived by the operator, no control manipulation is performed.

`[control_input_value]:` Format: Double. Range: any, but should be consistent with the control range of the controller specified by the ***action_name*** variable. This data filed must be provided when the ***action_type*** is 3074 (VACTION) or 3112 (VADDITIVE).

`[control_parameter]:` Format: String. Range: Must match a valid ***parameter_name***, listed in the "ControlPanel.txt" input file. Specifies the control panel indicator value that will be used to provide the control value for the ***action_name*** controller. This field shall only be included when the ***action_type*** is 3139 (VPARAMETER_CONTROL).

`lower_control_limit:` Format: Double. Range: any, but should be consistent with the control range of the controller specified by the ***action_name*** variable. Specifies the lower control limit for the ***action_name*** controller. Actions that would cause the controller position to decrease below the ***lower_control_limit*** are not performed.

upper_control_limit: Format: Double. Range: any, but should be consistent with the control range of the controller specified by the ***action_name*** variable. Specifies the upper control limit for the ***action_name*** controller. Actions that would cause the controller position to increase above the ***upper_control_limit*** are not performed.

`number_of_prerequsites:` Format: Integer. Range: ≥ 0. Specifies the number of prerequisites associated with the knowledge-based action. The ***number_of_prerequisites*** must exactly match the number of expectation units included with the knowledge-based action or an error will occur during input file processing or program execution. Similar to procedure step expectations, action prerequisites are modeled with a simple logic tree approach (Figure 17):

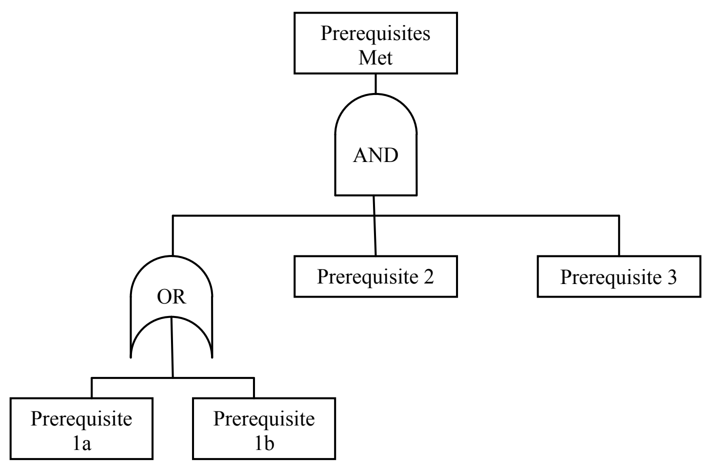


Figure 17, "Prerequisite Logic Tree"

In this example, the action prerequisites are met only if all of the following conditions are true: (1) either expectation 1a or 1b is true, (2) expectation 2 is true, and (3) expectation 3 is true. Procedure expectations are verified in sequential order until it is determined that either the action expectations are met or not met (e.g., once it is determined that the action expectation cannot be met, any remaining expectations will not be verified). For the purposes of counting the number of prerequisites, only prerequisite units connected with "AND" logic gates are counted. Therefore, this example has only three prerequisite conditions (since Prerequisites 1a and 1b are connected with an "OR" logic gate).

`item_name:` Format: String. Range: Must meet either of the following criteria: (1) match a valid ***parameter_name***, ***component_name***, or ***alarm_name*** listed in the "ControlPanel.txt" input file, or (2) be set to the reserved word **Mental_Belief**

or **Parameter_Difference**. If the ***item_name*** is set to a valid ***parameter_name***, ***component_name***, or ***alarm_name,*** the specified indicator is compared to an expected value or state. If the ***item_name*** is set to the reserved word **Mental_Belief**, the mental belief specified by ***parameter_1*** is compared to an expected state. If the ***item_name*** is set to the reserved word **Parameter_Difference**, the difference between ***parameter_1*** and ***parameter_2*** is compared to an expected value.

`[parameter_1]:` Format: String. Range: Must meet either of the following criteria: (1) if ***item_name*** is set to Mental_Belief, ***`parameter_1`*** must match a valid ***`mental_belief_name`*** listed in the "KB_OAT(ODM)_HardWired_Diagnosis.txt" input file, or (2) if the ***item_name*** is set to Parameter_Difference, ***`expect_parameter_1`*** must match a valid ***parameter_name*** listed in the "ControlPanel.txt" input file. This variable is only provided if the ***item_name*** is set to **Mental_Belief** or **Parameter_Difference**.

`[parameter_2]:` Format: String. Range: Must match a valid ***parameter_name*** listed in the "ControlPanel.txt" input file. This variable is only provided if the ***item_name*** is set to **Parameter_Difference**.

`[verification]:` Format: Integer. Range: Must be one of the following integer codes 3002 (VYES), 3003 (VNO), or 3045 (VNONE). When the operator is relying on the use of memorized information (set by the ***`information_branch_probability`*** in the "ActionTaker.txt" or "DecisionMaker.txt" input files), prerequisites may be evaluated using old information, particularly during dynamic situations. Although this may model real operator behavior in certain situations, there are times when operators would be expected to re-verify control panel indicators even when the indicator had been perceived earlier. In order to force the operator to perform this reverification, the ***verification*** variable can be set to 3002 (VYES). If the ***verification*** variable is set to 3003 (VNO) or 3045 (VNONE) the operator will not reverify the control panel indications and will always use previously perceived information (if available). If the operator has not previously perceived the indicator value or state (or if the information is deemed to be too old), the operator will reverify the information regardless of the value of the ***verification*** variable.

`expected_state:` Format: Integer. Range: Depending on the value of ***item_name*** variable and/or the type of control panel indicator used to evaluate the expectation, the ***expected_state*** should conform to the following guidelines:

expect_name **Reference**	**Integer code**	**Description**	**Comment**
Component	3022 3023 3098 3099	VON VOFF VOPEN VCLOSE	Integer code should be consistent with the ***threshold_type*** specified for the component in the "ControlPanel.txt" input file input file for the associated component

Alarm	3022 3023	VON VOFF	Alarm state is VON when alarm is activated
Mental Belief	3021 3045	VSUCCEED VNONE	All mental beliefs are initialized to the state VNONE at the beginning of the simulation.

The ***expected_state*** is only provided if the ***item_name*** is set to either (1) Mental_Belief, or (2) a valid alarm or component name listed in the "ControlPanel.txt" input file.

[relationship] : Format: Integer. Range: 3006 (VGT), 3007 (VGE), 3008 (VEQ), 3009 (VLE), 3010 (VLT), or 3032 (VBETWEEN). Specifies the type of comparison to be used to evaluate the prerequisite condition. The following comparison types may be used:

3006 (Integer code for "VGT"): If parameter referenced by ***item_name*** is greater than the ***value_1***, the prerequisite condition is satisfied.

3007 (Integer code for "VGE"): If parameter referenced by ***item_name*** is greater than or equal to the ***value_1***, the prerequisite condition is satisfied.

3008 (Integer code for "VEQ"): If parameter referenced by ***item_name*** is equal to the ***value_1***, the prerequisite condition is satisfied.

3009 (Integer code for "VLE"): If parameter referenced by ***item_name*** is less than or equal to the ***value_1***, the prerequisite condition is satisfied.

3010 (Integer code for "VLT"): If parameter referenced by ***item_name*** is less than the ***value_1***, the prerequisite condition is satisfied.

3032 (Integer code for "VBETWEEN"): If parameter referenced by ***item_name*** is between the ***value_1*** and the ***value_2***, the prerequisite condition is satisfied. ***value_1*** must be less than the value of ***value_2***.

[value_1]: Format: Double. Range: any, but should be consistent with range of the associated ***item_name*** parameter. Specifies the threshold value used to evaluate the action prerequisite.

[value_2]: Format: Double. Range: any, but should be consistent with range of the associated ***item_name*** parameter. Specifies the threshold value used to evaluate the action prerequisite. Although the ***value_2*** variable is only used to evaluate the expectation if the relationship is set to 3032 (VBETWEEN), a dummy value must be supplied for all other relationship values.

[logic_flag]: Format: Integer. Range: Should be one of the following integer codes: 3002 (VYES), 3003 (VNO), or 3045 (VNONE). The ***logic_flag*** is used to establish the Boolean relationship between successive prerequisite conditions. If the ***logic_flag*** is set to 3002 (VYES), the current prerequisite condition and the next prerequisite will be connected with "OR" gate logic to form one prerequisite unit. Prerequisite conditions connected with the logic_flag set to 3002 (VYES) count as a single prerequisite unit for the purposes of setting the ***number_of_prerequisites*** variable. When the VYES option is selected, satisfying any one of the associated prerequisite conditions will satisfy the entire prerequisite unit. When the logic_flag is set to 3003 (VNO) or 3045 (VNONE), the current prerequisite and the next prerequisite are treated as separate prerequisite units and are connected by "AND" gate logic. In this case, each prerequisite condition must be satisfied in order to satisfy the complete knowledge-based action prerequisite set. There is no limit on the number of individual prerequisites that can be connected with "OR" and "AND" gate logic. Thus, it is possible to create complex prerequisite requirements.

4. <u>Sample Input</u>

```
number_of_diagnoses     2

diagnosis_name          Mass_Imbalance_PZR_Low
diagnosis_priority      2
reset_delay_time        100.0 1.0   1.0
number_of_actions       2

action_name       X_PZR_PORV
action_priority   2
action_type       3074
control_input     0.0
lower_limit       0.0
upper_limit       1.0
number_of_prerequisites       1
PZR_Pressure  3045  0.0 1.0   2.0   3045  3009  2290.0  0.0 3045

action_name       X_SIAS
action_priority   3
action_type       3074
control_input     1.0
lower_limit       0.0
upper_limit       1.0
number_of_prerequisites 3
PZR_Level         3045  0.0   1.0   2.0   3045  3009  0.20    0.0
      3003
PZR_Pressure      3045  0.0   1.0   2.0   3045  3009  2150.0   0.0
      3003
Safety_Injection  3045  0.0   1.0   2.0   3023  3045

diagnosis_name          Mass_Imbalance_PZR_High
diagnosis_priority      5
reset_delay_time        200.0 1.0   1.0
number_of_actions       2

action_name       X_HPI_Pump_A
action_priority   2
action_type       3074
control_input     0.0
lower_limit       0.0
```

```
upper_limit       1.0
number_of_prerequisites 3
A_HPI_Pump_On     3045  1.0   3.0   2.0   3022  3003
Min_Sub_Cooling   3045  1.0   3.0   2.0   3045  3007  10.0  0.0
      3045
PZR_Level         3045  0.0   1.0   2.0   3045  3007  0.20  0.0
      3045

action_name       X_HPI_Pump_B
action_priority   3
action_type       3074
control_input     0.0
lower_limit       0.0
upper_limit       1.0
number_of_prerequisites 3
B_HPI_Pump_On     3045  1.0   3.0   2.0   3022  3045
Min_Sub_Cooling   3045  1.0   3.0   2.0   3045  3007  10.0  0.0
      3045
PZR_Level         3045  0.0   1.0   2.0   3045  3007  0.20  0.0
      3045
```

The above example provides knowledge-based actions for two imbalance diagnoses: (1) a low mass imbalance in the pressurizer and (2) a high mass imbalance in the pressurizer. The low mass pressurizer imbalance condition (i.e., low pressurizer water inventory) includes the following actions:

- Manually closing the pressurizer power operated relief valve (i.e., positioning the X_PZR_PORV controller to a control position of 0.0) provided that the pressurizer pressure is less than the PORV setpoint of 2290 psig. Because this action has a higher priority (i.e., a priority value closer to 1), it will be performed first.

- Manually actuating safety injection by positioning the X_SIAS controller to a control position of 1.0 provided that the pressurizer level is less than 20% and pressurizer pressure is less than 2150 psig, and safety injection has not already been actuated.

The high mass pressurizer imbalance condition (i.e., high pressurizer water inventory), specifies the following actions:

- Manually stop the A high pressure safety injection pump by positioning the X_HPI_Pump_A controller to a control position of 0.0 provided that the pump is on, the pressurizer water level is greater than 20%, and the minimum subcooling margin is greater than 10 F. This action has the higher priority and is performed first.

- Manually stop the B high pressure safety injection pump by positioning the X_HPI_Pump_B controller to a control position of 0.0 provided that the pump is on, the pressurizer water level is greater than 20%, and the minimum subcooling margin is greater than 10 F. This action has a lower priority and would be performed later during the event sequence.

General Description:

ADS-IDAC includes the capability to represent both the structure and content of many types of plant procedures. Procedure step execution follows the standard format of action execution followed by expectation verification. If the action expectations are not met, a mitigative action can be performed. Four general types of procedural actions can be executed: (1) changing the component operating mode (e.g., automatic vs. manual mode), (2) setting a specific control value for a component (e.g., throttling control valve to 50% open), (3) incrementing the control setting of a component (e.g., throttling open a control valve by an additional 10%), and (4) setting a control value based on a perceived parameter (e.g., setting the steam dump target pressure equal to the perceived main steam header pressure). These capabilities provide sufficient flexibility to realistically model all significant operator interactions with the plant model.

Generally, a written procedure is continued until the procedure is completed. However, the procedure flow may be interrupted by procedure transfers (which direct the crew to a different procedure), activation of an instinctive response action, or abandonment of the "Follow Written Procedure" strategy. Two types of procedure transfers can be modeled: (1) a permanent procedure transfer and (2) a temporary transfer to an auxiliary procedure followed by resumption of the initial procedure. An example of the first type of procedure transfer is the transfer from a general reactor trip procedure to a more specific emergency procedure (e.g., transfer from the Westinghouse E-0 to E-3 procedures during a steam generator tube rupture event). The second type of transfer supports implementation of functional recovery guidelines that are used to temporarily interrupt the current procedure to address a degraded condition.

Four types of event sequence branches can be generated during procedure execution: (1) mental procedure activation time branches, (2) action execution time branches, (3) action control value branches, and (4) step skipping. After a mental belief listed in the "KB_OAT(ODM)_HardWired_Diagnosis.txt" is activated, the associated memorized mental procedure is initiated after the activation time delay has elapsed. Mental procedure activation time branches allow the analyst to examine the impact of variations in the activation time delay. Action execution time branches enable multiple event sequence branches to be generated to model variations in the time taken by the control room crew in performing procedure actions. Action control value branches can be used to model variations in control inputs such as control valve positioning and the setting of control system target setpoints. Finally, procedure step skipping branches model the omissions of procedure actions based on the relevance of the step actions to the operator's situational assessment. The ADS-IDAC step skipping model is described in more detail in the "Procedures.txt" input file description.

Input Files:

MentalProcedureActivationTimeBranches.txt
ProcedureActionTimeBranches.txt
ProcedureControlValueBranches.txt
Procedures.txt
ZProcedure***_procedure_name_step_name***.txt

1. <u>Purpose</u>

The "MentalProcedureActivationTimeBranches.txt" input file allows multiple accident sequence branches to be generated when a memorized mental procedure is activated. Following activation of a mental belief, the operator will implement the associated mental procedure (if one is specified) after an activation time delay has elapsed. In order to capture the uncertainty and crew-to-crew variability associated with the time delay between activation of a mental belief and the execution of the associated memorized mental procedure, the activation delay is modeled with a three-parameter Weibull probability density distribution. When only one mental procedure activation time branch is generated, the activation time delay is equal to the mean value of Weibull distribution. If more than one activation time delay branch is generated, the probability distribution is partitioned into one or more segments and the time delay for each sequence branch is determined by the mean value over the associated partition. The partition boundaries are determined by dividing the probability range of the Weibull cumulative probability distribution function into a number of segments equal to the number of desired branches. If five activation time branches were to be generated, the cumulative probability distribution would be partitioned into five different segments (see Figure 18).

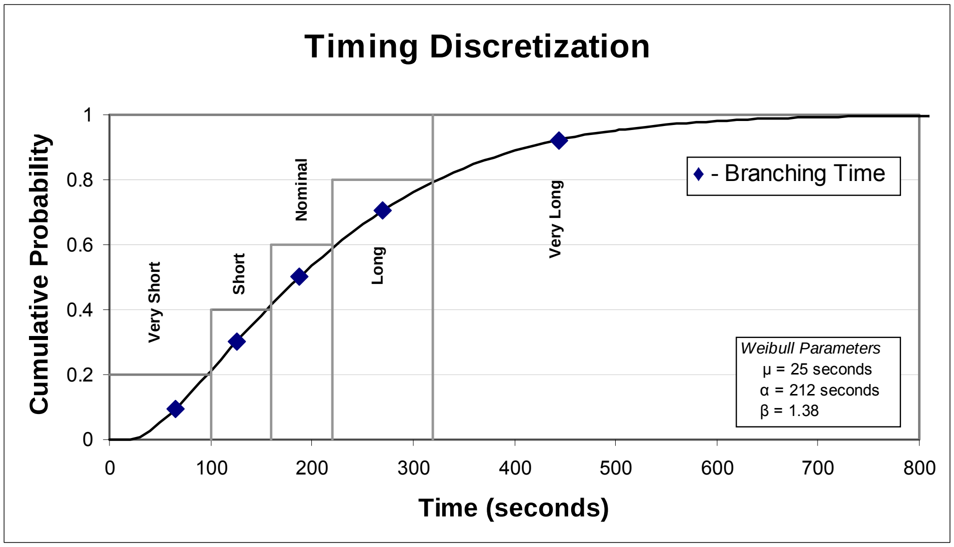


Figure 18, "Partitioned Weibull Time Distribution (Activation Branching)"

The activation time for each branch is the mean activation time over the associated partition. For example, the activation time for the first and second branch partitions are found as follows (Equation 10):

$$\bar{t}_{partition\ 1} = \frac{\int_{0.0}^{t_{1-2}} t_{delay}\ p(t_{delay})\,dt}{\int_{0.0}^{t_{1-2}} p(t_{delay})\,dt}$$

(Equation 10)

$$\bar{t}_{partition\ 2} = \frac{\int_{t_{1-2}}^{t_{2-3}} t_{delay}\, p(t_{delay})\,dt}{\int_{t_{1-2}}^{t_{2-3}} p(t_{delay})\,dt}$$

Where t_{1-2} and t_{2-3} correspond to the partition times between partions 1 and 2 and partitions 2 and 3, respectively. In this case, where F(t) is the cumulative probability density function, $F(t_{1-2}) = 0.2$ and $F(t_{2-3}) = 0.4$. This approach captures the probabilistic nature of the mental procedure activation time while maintaining the reproducibility of the analysis.

2. Input File Format

```
Instinctive_Response_Activation_Time_Branches    number_of_items
mental_belief_1         mental_procedure_1
      number_of_branches
.                       .                         .
mental_belief_1         mental_procedure_1
      number_of_branches
```

3. Input Description

number_of_items: Format: Integer. Range: ≥ 0. Specifies the number of mental procedure activation time branches included in the "MentalProcedureActivationTimeBranches.txt" input file. Branching events are read by the program in the order that they are listed in the input file. Therefore, if more than the specified ***number_of_items*** are described in the input file, only the first ***number_of_items*** are read.

Changing the ***number_of_items*** variable provides a convenient method for enabling and disabling branching events. The analyst can disable all branching events by setting the ***number_of_items*** variable to 0 (it is not necessary to delete each branching event from the file). If only one branching event will be activated, the desired event is simply moved to the beginning of the branching event list and the ***number_of_items*** variable is set to 1 (all other branching events will be ignored and need not be removed from the input file).

mental_belief: Format: String. Range: Must match a valid ***mental_belief_name*** in the "KB_OAT(ODM)_HardWired_Diagnosis.txt" input file. Specifies the mental belief the mental belief that activates the branching event.

mental_procedure: Format: String. Range: Must match a valid mental procedure ***procedure_name*** listed in the "Procedures.txt" input file. Mental procedures are identified by the ***procedure_name*** prefix "MPBG_".

number_of_branches: Format: Integer. Range: $\geq$ 1. Specifies the number of event sequence branches that will be generated.

4. Sample Input

```
Instinctive_Response_Activation_Time_Branches         2
SG_A_SGTR_Isolate_AFW   MPBG_SG_A_Isolate_AFW         3
SG_A_FWRV_Manual_Mode   MPBG_SG_A_FWRV_Manual_Mode    5
```

The sample input describes mental procedure activation time branching rules:

- When the "SG_A_SGTR_Isolate_AFW" mental belief is activated, three mental activation time branches are generated prior to initiation of the mental procedure "MPBG_SG_A_Isolate_AFW". Each time branch corresponds to a different partition of the activation time probability distribution. The first branch represents the shortest activation time and the third branch represents the longest activation time delay.

- When the "SG_A_FWRV_Manual_Mode" mental belief is activated, five timing branches are generated prior to initiation of the "MPBG_SG_A_FWRV_Manual_Mode" mental procedure. The first branch represents the shortest activation time and the fifth branch represents the longest activation time delay.

1. Purpose

The "ProcedureActionTimeBranches.txt" input file allows multiple accident sequence branches to be generated when a procedure action is executed. The ADS-IDAC procedure following model allows the analyst to specify the time taken to execute each step action. In order to capture the uncertainty and crew-to-crew variability associated with the time required to perform a proceduralized action, the time required to execute an action is modeled with a three-parameter Weibull probability density distribution. When only one action time branch is generated, the required action execution time is equal to the mean value of Weibull distribution. If more than one action execution time branch is generated, the probability distribution is partitioned into one or more segments and the time for each sequence branch is determined by the mean value over the associated partition. The partition boundaries are determined by dividing the probability range of the Weibull cumulative probability distribution function into a number of segments equal to the number of desired branches. For example, if five activation time branches were to be generated, the cumulative probability distribution would be partitioned into five different segments (see Figure 19).

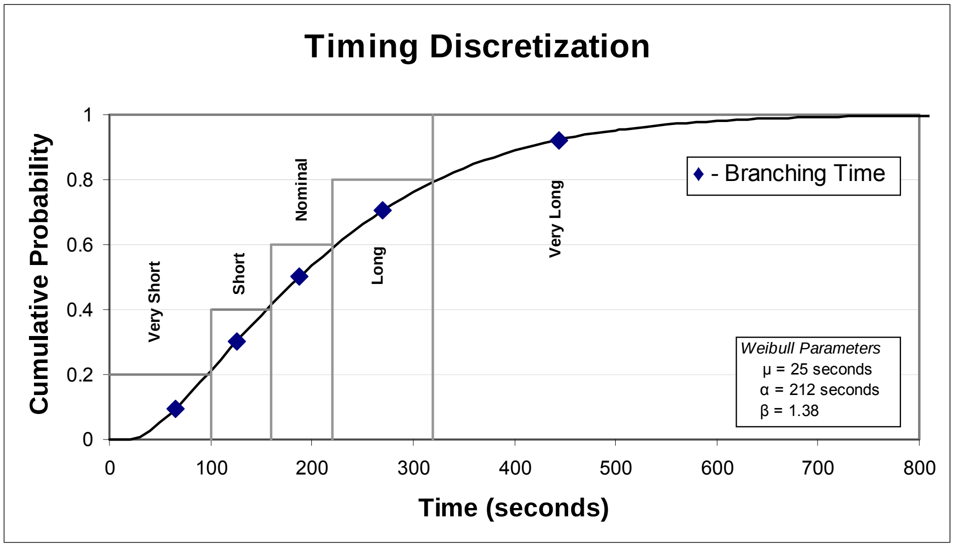


Figure 19, "Partitioned Weibull Time Distribution (Procedure Branching)"

The execution time for each branch is the mean time over the associated partition. For example, the activation time for the first and second branch partitions are found as follows (Equation 11):

$$\bar{t}_{partition\ 1} = \frac{\int_{0.0}^{t_{1-2}} t_{delay}\ p(t_{delay})\, dt}{\int_{0.0}^{t_{1-2}} p(t_{delay})\, dt}$$

(Equation 11)

$$\bar{t}_{partition\ 2} = \frac{\int_{t_{1-2}}^{t_{2-3}} t_{delay}\ p(t_{delay})\, dt}{\int_{t_{1-2}}^{t_{2-3}} p(t_{delay})\, dt}$$

Where $t_{1\text{-}2}$ and $t_{2\text{-}3}$ correspond to the partition times between partions 1 and 2 and partitions 2 and 3, respectively. In this case, where F(t) is the cumulative probability density function, $F(t_{1\text{-}2}) = 0.2$ and $F(t_{2\text{-}3}) = 0.4$. This approach captures the probabilistic nature of the action execution time while maintaining the reproducibility of the analysis.

2. Input File Format

```
Procedure_Action_Time_Branches             number_of_items
procedure_name_1   step_name_1   action_name_1
number_of_branches
.                  .             .             .
procedure_name_1   step_name_1   action_name_1
number_of_branches
```

3. Input Description

number_of_items: Format: Integer. Range: ≥ 0. Specifies the number of procedure action time branches included in the "ProcedureActionTimeBranches.txt" input file. Branching events are read by the program in the order that they are listed in the input file. Therefore, if more than the specified ***number_of_items*** are described in the input file, only the first ***number_of_items*** are read.

Changing the ***number_of_items*** variable provides a convenient method for enabling and disabling branching events. The analyst can disable all branching events by setting the ***number_of_items*** variable to 0 (it is not necessary to delete each branching event from the file). If only one branching event will be activated, the desired event is simply moved to the beginning of the branching event list and the ***number_of_items*** variable is set to 1 (all other branching events will be ignored and need not be removed from the input file).

procedure_name: Format: String. Range: Must match a valid procedure ***procedure_name*** listed in the "Procedures.txt" input file. Specifies the procedure associated with the branching rule.

step_name: Format: String. Range: Must match a valid procedure ***step_name*** listed in the "Procedures.txt" input file. Specifies the procedure step associated with the branching rule.

action_name: Format: String. Range: Must match a ***control_name*** listed in the "ControlPanel.txt" input file. If the ***control_name*** is not referenced by the associated procedure step, no branches will be generated (and no error will be generated).

number_of_branches: Format: Integer. Range: ≥ 1. Specifies the number of event sequence branches that will be generated.

4. Sample Input

```
Procedure_Action_Time_Branches      2
MPBG_SG_A_Increase_FWRV       Step_1            X_SG_A_FWRV 2
E_3                           Briefing_Hold_1   Time        3
```

This sample input defines two action time branching rules:

- When the action "X_SG_A_FWRV" is executed by procedure "MPBG_SG_A_Increase_FWRV", step "Step_1", two time branches will be generated to model variability in the specified action time. Execution of the action "X_SG_A_FWRV" by a different procedure step will not generate branching events.

- When the time delay action "Time" is executed by procedure "E_3", step "Briefing_Hold_1", three action time branches will be generated. This branching rule will generate three delay times – a short time, a nominal time, and a long time based on the time probability distribution specified for the action.

1. <u>Purpose</u>

For procedure step actions associated with a quantitative control input (e.g., opening a throttle valve to 10% open), two or more branches can be generated to explore the effect of variations in the control input. The "ProcedureControlValueBranches.txt"is used to specify when these quantitative control value sequence branching points will be generated and the specific control values to be used.

2. <u>Input File Format</u>

```
Procedure_Control_Value_Branches     number_of_items
procedure_name_1  step_name_1   action_name   number_of_branches
                  control_value_1   branch_probability_1
                  .                 .
                  control_value_m   branch_probability_m
.                 .                 .
.                 .                 .
procedure_name_n  step_name_n   action_name   number_of_branches
                  control_value_1   branch_probability_1
                  .                 .
                  control_value_m   branch_probability_m
```

3. <u>Input Description</u>

number_of_items: Format: Integer. Range: ≥ 0. Specifies the number of procedure action time branches included in the "ProcedureActionTimeBranches.txt" input file. Branching events are read by the program in the order that they are listed in the input file. Therefore, if more than the specified ***number_of_items*** are described in the input file, only the first ***number_of_items*** are read.

Changing the ***number_of_items*** variable provides a convenient method for enabling and disabling branching events. The analyst can disable all branching events by setting the ***number_of_items*** variable to 0 (it is not necessary to delete each branching event from the file). If only one branching event will be activated, the desired event is simply moved to the beginning of the branching event list and the ***number_of_items*** variable is set to 1 (all other branching events will be ignored and need not be removed from the input file).

procedure_name: Format: String. Range: Must match a valid procedure **procedure_name** listed in the "Procedures.txt" input file. Specifies the procedure associated with the branching rule.

step_name: Format: String. Range: Must match a valid procedure ***step_name*** listed in the "Procedures.txt" input file. Specifies the procedure step associated with the branching rule.

action_name: Format: String. Range: Must match a ***control_name*** listed in the "ControlPanel.txt" input file. If the ***control_name*** is not referenced by the associated procedure step, no branches will be generated (and no error will be generated).

number_of_branches: Format: Integer. Range: ≥ 1. Specifies the number of event sequence branches that will be generated.

control_value_i: Format: Double. Range: any – the ***control_value*** should be appropriate for the associated controller. Specifies the control value to be applied to the associated action.

branch_probability_i: Format: Double. Range: the branch probability should be within the range 0.0 – 1.0. If the total sum of all ***branch_probabilities*** for the branching rule do not sum to 1.0, an error message will be generated (but the branching rule will still be used if activated).

4. Sample Input

```
Procedure_Control_Value_Branches    2
FRG_H.1      Step_1.2    X_Watchdog_Timer  3
1200.0       0.25
1800.0       0.50
2400.0       0.25
FRG_H.1      Step_7.5    X_Stm_Dump        2
0.07  0.5
0.25  0.5
```

This sample input defines two control value branching events:

- When the action "X_Watchdog_Timer" is executed by procedure "FRG_H.1" step "Step_1.2", three control value branches are generated. The first branch sets the control value for "X_Watchdog_Timer" to 1200.0 seconds with a conditional branch probability of 0.25. The second branch sets the control value to 1800.0 seconds with a conditional probability of 0.5 and the third branch sets the control value to 2400.0 seconds with a probability of 0.25. The watchdog timer is used to trigger certain operator actions when the timer has elapsed to prevent excessive looping within a procedure.

- When the action "X_Stm_Dump" is executed by the procedure "FRG_H.1" step "Step_7.5", two control value branches are generated. The first branch sets the opening position of the steam dump valve (X_Stm_Dump) to 0.07

(7% open) with a probability of 0.5 and the second branch sets the valve opening position to 0.25 (25% open) with a probability of 0.5.

1. Purpose

a. General Overview

The “Procedures.txt” input file serves two main functions: (1) providing a master index list of all modeled procedure steps, and (2) specifying a number of critical variables needed to determine the probability of skipping a procedure step. The master index of procedure steps is used during input file processing in order to all identify the “ZProcedure***_procedure_name_step_name***.txt” input files provided for the project. The critical variables for the procedure step skipping model include static factors quantifying the procedural adherence tendencies for various types of procedure types and step objectives. The step skipping model is described in the remainder of this section.

b. Step Skipping Model

ADS-IDAC supports the modeling of omission of certain procedure actions in order model step skipping behavior. In order to provide adequate control over the simulation, step skipping behavior is limited to initial step actions and contingency “response not obtained” actions. The simulation approach requires that procedure steps be performed in sequence and that skipping behavior is applied at the sub step level. Although, ADS-IDAC cannot currently model the skipping of whole procedure sections, if the steps within a section are subject to the same dependent factors, the model can generate sequences where all section steps are skipped. The likelihood of skipping a sub-step is calculated by adjusting a base “skip step” probability by dynamic and static multipliers. These multipliers reflect procedural characteristics, the relevance of the action to the operator’s situational assessment, and the state of performance influencing factors.

Static Factors

Static factors refer to the properties of the procedure and are not expected to change during the accident event. In ADS-IDAC, static factors include procedure type, step objectives, and step complexity. In the U.S., quality assurance program requirements require that plant operators to specify the manner in which procedures are to be executed (ANS 3.2/ANSI N18.7, 1976). The methods used by operators to execute procedures can vary depending on the type of procedure. Routine procedural actions that are frequently repeated may not require the procedure to be present. Conversely, procedures covering infrequent or complex tasks should normally be present at the job site and followed. Six procedure types are considered: normal operating, alarm response, abnormal, emergency optimal recovery guidelines, emergency functional recovery guidelines, and mental (skill of the craft) procedures. Each procedure type is assigned a factor from 1 to 10 to reflect the procedural adherence tendencies of the operators (with high values indicating a lower adherence tendency).

The objectives of procedure steps may also affect the operator's adherence tendency. For example, steps that are clearly aligned with the high level objectives of a procedure are unlikely to be skipped while monitoring or verification activities might be more likely to be missed. ADS-IDAC uses the following five categories to group step objectives: monitoring, prerequisite, verification, objective-related, and diagnosis-related steps. Monitoring steps require the operator to periodically check the value of a parameter or condition while verification steps require the operator to ensure that an expected condition exists. Prerequisite steps support later actions but are not directly associated with the high level goals of the procedure. Objective-related steps are directly associated with the high level goals of the procedure. Diagnosis steps require the operators to assess the plant state and possibly transfer to a new procedure path. Similar to the procedure type, the analyst assigns a factor from 1 to 10 to reflect the operator's tendency to skip these various step types.

The complexity of the procedure step is also considered a static factor. Complexity can refer to the step structure, the type of action, and the presence of actions inside and outside of the control room. Similar to the static procedural factors, the static step complexity factor ranges from 1 to 10, with a higher value reflecting a greater tendency for action skipping. The three static factors (procedure type, step objective, and complexity) are multiplied together to provide an overall static factor (f_{static}) for step skipping.

Dynamic Factors

Two types of dynamic factors are used to adjust the basic step skipping probability: (1) performance influencing factors, and (2) the relevance of the action to the operator's situational assessment. Because high time pressure may influence an operator's tendency to skip procedure steps, a time constraint loading performance influencing factor (PIF) is included in the step skipping model. The time constraint load PIF varies in the range of 1 to 10, with a higher value indicating increased time pressure. The relevance of an action to the operator's situational assessment is determined by comparing the functions of the component references by the action to the output from the diagnostic engine. The plant component functional map (Section 2.3) specifies all functions supported by a component. Further, each function is directly associated to an imbalance event included in the diagnostic engine. Based on information perceived by the operator, the diagnostic engine calculates a membership value, d, for each imbalance diagnosis. A relevance score for each component action, R_{action}, is then calculated from the following equation (12):

$$R_{action} = 10^{1-2d} \qquad \text{(Equation 12)}$$

where R_{action} is the relevance factor and d is the maximum membership value of all functional imbalances associated with the action. Because d varies from 0.0 to 1.0, the relevance factor, R, varies from 0.1 for highly relevant actions to 10.0 for irrelevant actions. Since the amount and accuracy of plant data perceived by the operator changes over time, the relevance factor is a dynamic quantity. An operator with an accurate situational assessment will be less likely to skip pertinent actions, while an operator with a poor situational assessment may skip important steps. Actions that are not associated with a specific component (such as procedure transfers) are assigned a relevance factor of 1.0. The action relevance factor (R_{action}) is multiplied by the time constraint load PIF to yield the overall dynamic factor ($f_{dynamic}$).

Calculating Overview Skip Probability

Based on the static and dynamic step factors, an adjusted step skipping probability is calculated using the following equation (13):

$$P_{skip} = \frac{P_{base} f_{static} f_{dynamic}}{\left[P_{base}\left(f_{static} f_{dynamic} - 1\right) + 1\right]} \qquad \text{(Equation 13)}$$

where P_{base} is the basic step skipping probability and P_{skip} is the adjusted probability. The dynamic calculation of the step skipping probability provides a number of advantages, including: (1) the ability to consider procedure type, step intent, and step complexity, (2) the influence of time pressure, and (3) the ability to link step skipping tendencies to the operator situational assessment through the relevance factor.

2. <u>Input File Format</u>

```
Step_Transfer_Time                              step_transfer_time
Procedure_Transfer_Time                         procedure_transfer_time

normal_procedure_multiplier                     skip_multiplier_normal
abnormal_procedure_multiplier
      skip_multiplier_abnormal
alarm_response_procedure_multiplier             skip_multiplier_alarm
optimal_recovery_procedure_multiplier           skip_multiplier_EOP
functional_recovery_procedure_multiplier        skip_multiplier_FRG
mental_procedure_multiplier                     skip_multiplier_mental

verification_step_multiplier
      skip_multiplier_verification
monitoring_step_multiplier
      skip_multiplier_monitoring
prerequisite_step_multiplier
      skip_multiplier_prerequisite
objective_step_multiplier
      skip_multiplier_objective
```

```
diagnosis_step_multiplier
      skip_multiplier_diagnosis

Coded_Steps       number_of_coded_steps
procedure_name_1        step_name_1
.                       .
procedure_name_n        step_name_n
```

3. Input Description

step_transfer_time: Format: Double. Range: ≥ 0.0. Specifies the nominal time delay to transition between steps within the same procedure.

procedure_transfer_time: Format: Double. Range: ≥ 0.0. Specifies the nominal time delay to transition between different procedures.

skip_multiplier_normal: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the step skipping multiplier for normal operating procedures. Normal procedures typically include routine power changes and routine plant evolutions. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of plant procedures – a higher value implies a greater likelihood of skipping the procedure step.

skip_multiplier_abnormal: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the step skipping multiplier for abnormal operating procedures. Abnormal procedures are typically used to address non-routine events that do not constitute emergency or accident situations. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of plant procedures – a higher value implies a greater likelihood of skipping the procedure step.

skip_multiplier_alarm: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the step skipping multiplier for alarm response procedures. Alarm response procedures are used to guide operator follow up actions after a control panel alarm is activated. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of plant procedures – a higher value implies a greater likelihood of skipping the procedure step.

skip_multiplier_EOP: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the step skipping multiplier for emergency operating procedures. Emergency operating procedures (EOPs) are used to mitigate accident conditions. EOPs are also known as "optimal recovery guidelines". The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of plant procedures – a higher value implies a greater likelihood of skipping the procedure step.

skip_multiplier_FRG: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the step skipping multiplier for functional recovery guidelines (FRGs). FRGs are used to address degradations of critical safety functions such as inventory control, core shutdown and cooling, and fission product containment. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of plant procedures – a higher value implies a greater likelihood of skipping the procedure step.

skip_multiplier_mental: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the step skipping multiplier for memorized mental procedures. Memorized mental procedures are used to model skill-based actions carried out by the operators without reference to written procedures. these actions typically fall into the broad category of skill-of-the-craft activities. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of plant procedures – a higher value implies a greater likelihood of skipping the procedure step.

skip_multiplier_verification: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the skip multiplier for proceduralized actions that perform verification functions. Verification functions include checking the status of parameters and components where the operator does not normally expect to perform recovery actions. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of procedure steps – a higher value implies a greater likelihood of skipping the procedure step.

skip_multiplier_monitoring: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the skip multiplier for proceduralized actions that perform monitoring functions. Monitoring functions are generally associated with steps where an operator is required to observe the status of a parameter or component while performing other actions in parallel. Monitoring also includes observing the status of a changing parameter in order to initiate action when a threshold value is reached. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of procedure steps – a higher value implies a greater likelihood of skipping the procedure step.

skip_multiplier_prerequisite: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the skip multiplier for proceduralized actions that perform verification functions. Prerequisite functions refer to actions that do not directly address the cause or symptoms of an ongoing event, but are needed to support later activities or prevent undesirable consequences of planned actions. For example, blocking the low pressure safety injection actuation prior to reactor coolant system depressurization would be considered a prerequisite action. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of procedure steps – a higher value implies a greater likelihood of skipping procedure steps.

skip_multiplier_objective: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the skip multiplier for proceduralized actions that perform verification functions. Objective functions directly address the cause or symptoms of an ongoing event. These actions are usually central to the operator's understanding of the overall procedure goals and are less likely to be skipped. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of procedure steps – a higher value implies a greater likelihood of skipping procedure steps.

skip_multiplier_diagnosis: Format: Double. Range: > 0.0, but should normally fall within the range of 1.0 – 10.0. Specifies the skip multiplier for proceduralized actions that perform verification functions. Diagnosis functions involve the identification of the root cause(s) of an abnormal or emergency event. Diagnosis activities are generally focused on the identification of a specific failed component or system so that mitigative actions can be performed. The skip multiplier is used to reflect the crew's procedural adherence tendencies for various types of procedure steps – a higher value implies a greater likelihood of skipping procedure steps.

number_of_coded_steps: Format: Integer. Range: ≥ 0. Specifies the number of procedure steps included in the procedure index list. Procedure step names are read by the program in the order that they are listed in the input file. Therefore, if more than the specified ***number_of_coded_steps*** are listed in the input file, only the first ***number_of_coded_steps*** are read.

procedure_name_i: Format: String. Range: Must be associated with a valid "ZProcedure_***procedure_name_step_name***.txt" input file. Spaces must not be used in the ***procedure_name*** (use the underbar character ("_") rather than a space when needed).Because mental procedures and functional recovery guidelines require special handling, the following procedure name prefixes are reserved to identify these procedure types:

"FRG_" – Functional Recovery Guideline
"MPBG_" – Mental Procedure

It is also recommended (though not required) that the following ***procedure_name*** prefixes be used:

"ECA_" - Emergency Contingency Actions,
"E_" - Emergency Operating Procedures
"ES_" - Emergency Supplemental Procedure

step_name_i: Format: String. Range: Must be associated with a valid "ZProcedure_***procedure_name_step_name***.txt" input file. Spaces must not be used (use the underbar character ("_") rather than a space when needed

4. Sample Input

```
Step_Transfer_Time                          0.5
Procedure_Transfer_Time                     5.0

normal_procedure_multiplier                 2.0
abnormal_procedure_multiplier               1.0
alarm_response_procedure_multiplier         5.0
optimal_recovery_procedure_multiplier       1.0
functional_recovery_procedure_multiplier    5.0
mental_procedure_multiplier                 10.0

verification_step_multiplier                5.0
monitoring_step_multiplier                  7.0
prerequisite_step_multiplier                3.0
objective_step_multiplier                   1.0
diagnosis_step_multiplier                   3.0

Coded_Steps       3
E_0   Step_1
E_0   Step_2
E_0   Step_3
```

The above sample provides profiling and indexing information needed to support procedure following. Several features of note include:

- The step transfer time is significantly shorter than the procedure transfer time. This reflects the greater ease crews would have in transitioning between steps within the same procedure compared to starting a new procedure.
- The step skipping multipliers reflect several crew important crew tendencies. These include the increased likelihood of skipping steps in mental procedures compared to written procedures and the increased procedural adherence tendency for EOPs compared to normal operating procedures.
- Three procedure steps are indexed in the sample input file, E-0, steps 1, 2, and 3. Each of these procedure steps should have an associated "ZProcedure***_procedure_name_step_name***.txt" input file.

1. Purpose

The "ZProcedure_***procedure_name_step_name***.txt" input file provides the detailed instructions to be followed when executing a procedure step. ADS-IDAC utilizes a standardized format for proceduralized actions that includes an action, followed by the verification of expectations to normally should be observed after the action is completed. If the expectations are not met, a mitigative action is executed. A procedural actions fall into five main categories: (1) changing the component operating mode (e.g., automatic vs. manual mode), (2) setting a specific control value for a component (e.g., throttling control valve to 50% open), (3) incrementing the control setting of a component (e.g., throttling open a control valve by an additional 10%), (4) setting a control value based on a perceived parameter (e.g., setting the steam dump target pressure equal to the perceived main steam header pressure), and (5) simple time delays. The first four action types actively change the status of a reactor plant component or system. The last action type is intended to model the time taken by operators to perform activities that are not included in the ADS-IDAC model. For example, activation of the emergency plan or alignment of equipment not included in the RELAP thermal hydraulic model is simulated by a simple time delay when the appropriate step is reached. Action expectations are modeled with a simple logic tree approach (Figure 20):

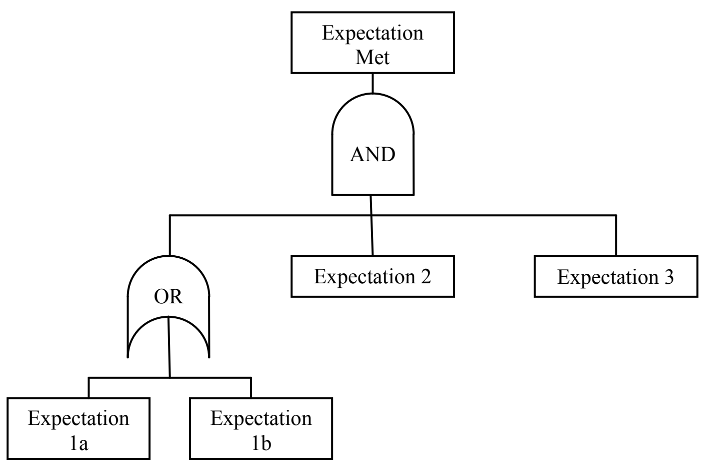


Figure 20, "Expectation Logic Structure"

In this example, the action expectation is met only if all of the following conditions are true: (1) either expectation 1a or 1b is true, (2) expectation 2 is true, and (3) expectation 3 is true. Procedure expectations are verified in sequential order until it is determined that either the action expectations are met or not met (e.g., once it is determined that the action expectation cannot be met, any remaining expectations will not be verified). If the action expectations are not met, a mitigative (or non-response) action is performed (if one is specified). In addition to the capability of performing all the same functions as the initial procedure step action, mitigative actions[11] can also be used to transfer to a different procedure step or terminate the accident sequence. Once expectations are verified and any applicable mitigative action performed, the procedure flow will progress to the next action step within the same procedure step (each procedure step can contain multiple possible action/expectation/mitigation units) or transition to the next procedure step. If the next procedure step is not specified, the procedure following strategy is terminated.

2. Input File Format

```
procedure_name          step_name   "step_description"
procedure_type          step_type    step_complexity
      double
number_of_substeps

action_name_1      [action_type]
[parameter_name    parameter_type    scan_priority]
minimum_time       weibull_alpha     weibull_beta
[control_value]    [control_parameter]
skip_step_alpha    skip_step_beta

      number_of_expectations
      expect_name_1    [expect_parameter_1
            expect_parameter_2]
      [verification}
      [minimum_time     weibull_alpha          weibull_beta]
      [threshold]
      [expected_state]   [relationship   expect_value_1
      expect_value_2]
      logic_flag
      .                  .
      .                  .
      expect_name_n     [expect_parameter_1
            expect_parameter_2]
      [verification}
      [minimum_time     weibull_alpha          weibull_beta]
      [threshold]
      [expected_state]   [relationship   expect_value_1
      expect_value_2]
      logic_flag

non_response_action_type_1
[procedure_name]                  [step_name]
[nonresponse_action_name_1]
[nonresponse_indicator_type]      [nonresponse_scan_priority]
```

[11] Mitigative actions are also referred to as contingency actions or non-response actions. These three terms are used interchangeable in this manual.

```
[minimum_time]    [weibull_alpha]   [weibull_beta]
[nonresponse_control_value]
      [nonresponse_control_parameter]
[skip_nonresponse_alpha]          [skip_nonresponse_beta]

.
.
.
[additional step actions...]

next_procedure_name     next_step_name
```

3. Input Description

procedure_name: Format: String. Range: Must be associated with a valid ***procedure_name*** and ***step_name*** listed in the "Procedures.txt" input file. Spaces must not be used (use the underbar character ("_") rather than a space when needed). Because mental procedures and functional recovery guidelines require special handling, the following procedure name prefixes are reserved to identify these procedure types:

"FRG_" – Functional Recovery Guideline
"MPBG_" – Mental Procedure

It is also recommended (though not required) that the following ***procedure_name*** prefixes be used:

"ECA_" - Emergency Contingency Actions,
"E_" - Emergency Operating Procedures
"ES_" - Emergency Supplemental Procedure

step_name: Format: String. Range: Must be associated with a valid ***procedure_name*** and ***step_name*** listed in the "Procedures.txt" input file. Spaces must not be used (use the underbar character ("_") rather than a space when needed).

"step_description": Format: String. Range: any. Entry must be delineated by quotation marks (""). Provides a descriptive title for the procedure step.

procedure_type: Format: Integer. Range: Must be one of the following integer codes:

Integer Code	Variable Name	Procedure Type
3151	VNOP	Normal Operating Procedure
3152	VAOP	Abnormal Procedure
3153	VEOP	Emergency Operating Procedure
3154	VFRG	Functional Recovery Guideline
3155	VECA	Emergency Contingency Action
3156	VMENTAL_PROCEDURE	Memorized Procedure

The ***procedure_type*** variable specifies the category of the procedure and is used to calculate the step skipping probability. The skip probability multiplier for these procedure types is specified in the "Procedures.txt" input file.

step_type: Format: Integer. Range: Must be one of the following integer codes:

Integer Code	Variable Name	Step Type
3161	VOBJECTIVE_RELATED	Objective related
3162	VPREREQUISITE_ACTION	Prerequisite
3163	VMONITORING_STEP	Monitoring
3164	VVERIFICATION_STEP	Verification
3165	VDIAGNOSIS_STEP	Diagnostic

The ***step_type*** variable specifies the category of the procedure and is used to calculate the step skipping probability. The skip probability multiplier for these step types is specified in the "Procedures.txt" input file.

step_complexity: Format: Integer. Range: > 0, but should normally fall within the range of 0.1 – 10.0. Specifies the step complexity multiplier used to calculate the step skipping probability. Steps consisting of multiple substeps, actions performed from multiple locations, or convoluted structure may have a higher likelihood for being skipped. In general, this parameter should be set to 1 (neutral complexity impact). A value of the ***step_complexity*** > 1.0 will increase the likelihood of skipping the step, while a value < 1.0 will decrease the likelihood of skipping the step.

double: Format: Double. Range: any. This is input variable is reserved for future use. Enter a dummy value (e.g., 1.0) for this parameter.

number_of_substeps: Format: Integer. Range: $\geq$ 0. Specifies the number of procedure substeps listed in the input file. A substep consists of an action, a set of expectations, and a contingency action that is performed if the expectations are not met. The ***number_of_substeps*** must exactly match the number of listed substeps in the input file or an error will occur during input file processing or program execution. If the number of substeps is 0, the substep action is a simple procedure step transfer and no other data fields shall be provided except for the ***next_procedure_name*** and ***next_step_name***.

action_name: Format: String. Range: Varies depending on value of the variable ***action_type*** and the use of special reserved words. In conjunction with the ***action_type*** variable, specifies the type of control panel manipulation to be performed by the action. Two special reserved values may be used for the ***action_name***:

Reserved Value	Description	Comments
0	Procedure transfer	No other data fields shall be provided in the remainder of the input file except for the ***next_procedure_name*** and ***next_step_name***
SCAN	Add control panel indicator to the operator scan queue	The ***parameter_name***, ***parameter_type***, and ***scan_prioirity*** variables must be provided. The ***action_type*** and ***control_value*** shall not be included.

If the ***action_type*** is VADDITIVE (integer code 3112), VACTION (integer code 3074), or VPARAMETER_CONTROL (integer code 3139), the ***action_name*** specifies the control to be manipulated and must refer to a valid controller listed in the "ControlPanel.txt" input file. If the ***action_name*** will initiate a maneuvering action, the ***action_name*** must include the prefix "MANEUVER_" and should reference a valid maneuvering action listed in the "KB_OAT(ODM)_Maneuvering_Actions.txt" input file.

[action_type]: Format: Integer. Range: Should refer to one of the following valid integer codes:

Integer Code	Description	Comments
3074	VACTION	Simple controller manipulation. The ***action_name*** controller will be positioned to the value specified by ***control_value***
3109	VVERIFY	No active control manipulation is performed. This action_type is used to model a time delay. A dummy ***control_value*** must still be provided.
3112	VADDITIVE	The ***action_name*** controller control value will be incremented by the amount specified by the ***control_value*** variable. The control value is bounded by the values for fully open/on and fully closed/off provided for the action_name control in the "ControlPanel.txt" input file.

Integer Code	Description	Comments
3139	VPARAMETER_CONTROL	The ***action_name*** controller will be positioned to the perceived value of the ***control_parameter*** indicator. If the ***contorl_parameter*** value has not been perceived by the operator, no control manipulation is performed.

The action_type is not provided when the action_name is either "SCAN" or "0".

[parameter_name]: Format: String. Range: Must match a valid ***parameter_name***, ***component_name***, or ***alarm_name*** listed in the "ControlPanel.txt" input file. Specifies the control panel indicator to be added to the operator's control panel scan queue. This field is only applicable when the ***action_type*** is set to SCAN.

[parameter_type]: Format: Integer. Range: 1, 2, or 3. Specifies the type of parameter represented by the ***parameter_name*** variable. The operator scan queue contains three sub-queue lists: control panel parameters, components, and alarms. The ***parameter_type*** is used to place the parameter_name in the correct scan queue. This field is only applicable when the ***action_type*** is set to SCAN. The following options are available:

parameter_type Value	Description	Comments
1	VALARM_STATE	Control panel alarm
2	VCOMPONENT_STATE	Control panel component status indicator
3	VPARAMETER_VALUE	Control panel parameter value indicator

[scan_priority]: Format: Integer. Range: $\geq$ 1. Specifies the initial priority level applied to the ***parameter_name*** added to the operator scan queue. A lower ***scan_priority*** value designates a higher priority control panel indicator (i.e., the highest priority items are designated with a priority level of 1). Because the priority level can decay over time, specifying a lower ***scan_priority*** value will increase the residence time of the alarm on the scan queue list. This field is only applicable when the ***action_type*** is set to SCAN.

minimum_time: Format: double. Range: > 0.0. In conjunction with the ***weibull_alpha*** and ***weibull_beta*** variables, specifies the time taken to perform the procedure substep action, evaluate substep expectations, or perform non-response actions. In order to capture the uncertainty associated with these times, a three parameter Weibull probability distribution is used. The Weibull distribution is given by the following equation (14):

$$F(t) = 1 - \exp\left[-\left(\frac{t-u}{\alpha}\right)^{\beta}\right]$$

(Equation 14)

Where:

minimum_time = u (minimum time), seconds
weibull_alpha = α parameter
weibull_beta = β parameter

This parameter must be provided for the following cases:

- Actions: Data field is required when the ***action_type*** is 3074 (VACTION), 3109 (VVERIFY), 3112 (VADDITIVE), or 3139 (VPARAMETER_CONTROL); or (2) the ***action_name*** SCAN is used. This parameter should not be entered for other action types.

- Expectations: Data field is required for all expectation types except when the ***expect_name*** is set to the reserved word Mental_Belief

- Nonresponse Actions: Data field is required when the ***nonresponse_action_type*** is set to 3074 (VACTION), 3112 (VADDITIVE), 3139 (VPARAMETER_CONTROL), or 3140 (VPARAMETER_SCAN). This parameter should not be entered for other nonresponse action types.

weibull_alpha: Format: double. Range: > 0.0. In conjunction with the ***minimum_time*** and ***weibull_beta*** variables, specifies the time taken to perform the procedure substep action. This parameter must be provided for the following cases:

- Actions: Data field is required when the ***action_type*** is 3074 (VACTION), 3109 (VVERIFY), 3112 (VADDITIVE), or 3139 (VPARAMETER_CONTROL); or (2) the ***action_name*** SCAN is used. This parameter should not be entered for other action types.

- Expectations: Data field is required for all expectation types except when the ***expect_name*** is set to the reserved word Mental_Belief

- Nonresponse Actions: Data field is required when the ***nonresponse_action_type*** is set to 3074 (VACTION), 3112 (VADDITIVE), 3139 (VPARAMETER_CONTROL), or 3140

(VPARAMETER_SCAN). This parameter should not be entered for other nonresponse action types.

weibull_beta: Format: double. Range: > 0.0. In conjunction with the ***minimum_time*** and ***weibull_alpha*** variables, specifies the time taken to perform the procedure substep action. This parameter must be provided for the following cases:

- Actions: Data field is required when the ***action_type*** is 3074 (VACTION), 3109 (VVERIFY), 3112 (VADDITIVE), or 3139 (VPARAMETER_CONTROL); or (2) the ***action_name*** SCAN is used. This parameter should not be entered for other action types.

- Expectations: Data field is required for all expectation types except when the ***expect_name*** is set to the reserved word Mental_Belief

- Nonresponse Actions: Data field is required when the ***nonresponse_action_type*** is set to 3074 (VACTION), 3112 (VADDITIVE), 3139 (VPARAMETER_CONTROL), or 3140 (VPARAMETER_SCAN). This parameter should not be entered for other nonresponse action types.

[control_value]: Format: Double. Range: any, but should be consistent with the control range of the controller specified by the ***action_name*** variable. This data filed must be provided when the ***action_type*** is 3074 (VACTION) , 3109 (VVERIFY), or 3112 (VADDITIVE). When the ***action_type*** is of type VVERIFY, only a dummy ***control_value*** need be provided (since no control manipulation is actually performed).

[control_parameter]: Format: String. Range: Must match a valid ***parameter_name***, listed in the "ControlPanel.txt" input file. Specifies the control panel indicator value that will be used to provide the control value for the ***action_name*** controller. This field shall only be included when the ***action_type*** is 3139 (VPARAMETER_CONTROL).

skip_step_alpha: Format: Double. Range: > 0.0. Specifies the α parameter in the procedure step skipping probability distribution. To capture the uncertainty associated with skipping procedure steps, the Beta Distribution is used to model the base step skipping probability (Equation 15):

$$p(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{(\alpha-1)} (1-x)^{(\beta-1)}$$

$$\overline{p} = \frac{\alpha}{\alpha+\beta} \quad (\textit{mean value})$$

(Equation 15)

Two parameters must be specified, the alpha parameter (α) and the beta parameter (β). During a simulation run, the step skipping probability is determined by a Monte Carlo sample of the Beta distribution. To improve the reproducibility of simulation results, it is recommended that the α and β parameters be selected to minimize the variance of the failure distribution. For example, the failure distributions p(x, α = 1. β = 1.) and p(x, α = 100. β = 100.) both have the same mean value (0.5), but the variance for the latter case is more than an order of magnitude lower. A smaller variance will yield more reproducible simulation results. As described in the "Procedures.txt" input file description, the base step skipping probability is modified by several multiplicative factors to account for status and dynamic effects. This parameter must be provided when: (1) the ***action_type*** is 3074 (VACTION), 3109 (VVERIFY), 3112 (VADDITIVE), or 3139 (VPARAMETER_CONTROL); or (2) the ***action_name*** SCAN is used.

skip_step_beta: Format: Double. Range: > 0.0. Specifies the β parameter in the base step skipping probability distribution. As described in the "Procedures.txt" input file description, the base step skipping probability is modified by several multiplicative factors to account for status and dynamic effects. This parameter must be provided when: (1) the ***action_type*** is 3074 (VACTION), 3109 (VVERIFY), 3112 (VADDITIVE), or 3139 (VPARAMETER_CONTROL); or (2) the ***action_name*** SCAN is used.

number_of_expectations: Format: Integer. Range: ≥ 0. Specifies the number of expectations associated with the substep action. The ***number_of_expectations*** must exactly match the number of expectation units included with the substep or an error will occur during input file processing or program execution. A single expectation unit is considered to be a complete block of verifications connected by "OR" logic. For example, in the expectation example provided in Figure 20, Expectation 1a and 1b, together, would be considered a single expectation unit, and the example would consist of a total of three expectation units.

If the ***number_of_expectations*** is set to 0, the ***non_response_action_type*** should be set to 3045 (VNONE), and no other data fields shall be provided in the remainder of the input file except for the ***next_procedure_name*** and ***next_step_name.***

[expect_name]: Format: String. Range: Must meet either of the following criteria: (1) match a valid ***parameter_name***, ***component_name***, or ***alarm_name*** listed in the "ControlPanel.txt" input file, or (2) be set to the reserved word **Mental_Belief** or **Parameter_Difference**. If the ***expect_name*** is set to a valid ***parameter_name***, ***component_name***, or ***alarm_name,*** the specified indicator is compared to an expected value or state. If the ***expect_name*** is set to the reserved word **Mental_Belief**, the mental belief specified by ***expect_parameter_1*** is compared to an expected state. If the ***expect_name*** is set to the reserved word **Parameter_Difference**, the difference between ***expect_parameter_1*** and ***expect_parameter_2*** is compared to an expected value.

[expect_parameter_1]: Format: String. Range: Must meet either of the following criteria: (1) if ***expect_name*** if set to Mental_Belief, ***expect_parameter_1*** must match a valid ***mental_belief_name*** listed in the "KB_OAT(ODM)_HardWired_Diagnosis.txt" input file, or (2) if the ***expect_name*** is set to Parameter_Difference, ***expect_parameter_1*** must match a valid ***parameter_name*** listed in the "ControlPanel.txt" input file. This variable is only provided if the expect_name is set to **Mental_Belief** or **Parameter_Difference**.

[expect_parameter_2]: Format: String. Range: Must match a valid ***parameter_name*** listed in the "ControlPanel.txt" input file. This variable is only provided if the expect_name is set to **Parameter_Difference**.

[verification]: Format: Integer. Range: Must be one of the following integer codes 3002 (VYES), 3003 (VNO), or 3045 (VNONE). When the operator is relying on the use of memorized information (set by the ***information_branch_probability*** in the "ActionTaker.txt" or "DecisionMaker.txt" input files) , expectations may be evaluated using old information, particularly during dynamic situations. Although this may model real operator behavior in certain situations, there are times when operators would be expected to re-verify control panel indicators even when the indicator had been perceived earlier. In order to force the operator to perform this reverification, the ***verification*** variable can be set to 3002 (VYES). If the ***verification*** variable is set to 3003 (VNO) or 3045 (VNONE) the operator will not reverify the control panel indications and will always use previously perceived information (if available). If the operator has not previously perceived the indicator value or state (or if the information is deemed to be too old), the operator will reverify the information regardless of the value of the ***verification*** variable. Setting ***verification*** to VYES is particularly useful when the procedure requires looping until a parameter reaches a threshold value – without continual reverification excessive looping may occur.

[threshold]: Format: Integer. Range: any, but should be consistent with range of the associated parameter. Specifies the threshold value used to evaluate a parameter difference. The expectation is satisfied in ***expect_parameter_1*** – ***expect_parameter_2*** is greater than the threshold value.

[expected_state]: Format: Integer. Range: Depending on the value of ***expect_name*** variable and/or the type of control panel indicator used to evaluate the expectation, the ***expected_state*** should conform to the following guidelines:

expect_name Reference	Integer code	Description	Comment
Component	3022 3023 3098 3099	VON VOFF VOPEN VCLOSE	Integer code should be consistent with the ***threshold_type*** specified for the component in the "ControlPanel.txt" input file input

			file for the associated component
Alarm	3022 3023	VON VOFF	Alarm state is VON when alarm is activated
Parameter	Any (3045 preferred)	n/a (Dummy Value)	Expected state not currently implemented for parameter value expectations – a dummy integer value should be specified (e.g., 3045)
Mental Belief	3021 3045	VSUCCEED VNONE	All mental beliefs are initialized to the state VNONE at the beginning of the simulation.

The ***expected_state*** is only provided if the ***expect_name*** is set to either (1) "Mental_Belief", or (2) a valid alarm, parameter, or component name.

`[relationship]:` Format: Integer. Range: 3006 (VGT), 3007 (VGE), 3008 (VEQ), 3009 (VLE), 3010 (VLT), or 3032 (VBETWEEN). Specifies the type of comparison used to evaluate the prerequisite condition. The following comparison types may be used:

3006 (Integer code for "VGT"): If parameter referenced by ***expect_name*** is greater than the ***expect_value_1***, the expectation is satisfied.

3007 (Integer code for "VGE"): If parameter referenced by ***expect_name*** is greater than or equal to the ***expect_value_1***, the expectation is satisfied.

3008 (Integer code for "VEQ"): If parameter referenced by ***expect_name*** is equal to the ***expect_value_1***, the expectation is satisfied. Because of rounding and data storage errors associated with real numbers, the VEQ condition should be used with care.

3009 (Integer code for "VLE"): If parameter referenced by ***expect_name*** is less than or equal to the ***expect_value_1***, the expectation is satisfied.

3010 (Integer code for "VLT"): If parameter referenced by ***expect_name*** is less than the ***expect_value_1***, the expectation is satisfied.

3032 (Integer code for "VBETWEEN"): If parameter referenced by ***expect_name*** is between the ***expect_value_1*** and the ***expect_value_2***, the expectation is satisfied. ***expect_value_1*** must be less than the value of ***expect_value_2***.

`[expect_value_1]:` Format: Double. Range: any, but should be consistent with range of the associated parameter. Specifies the threshold value used to evaluate the expectation.

`[expect_value_2]:` Format: Double. Range: any, but should be consistent with range of the associated parameter. Specifies the threshold value used to evaluate the expectation. Although the ***expect_value_2*** variable is only used to evaluate the expectation if the relationship is set to 3032 (VBETWEEN), a dummy value must be supplied for all other relationship values.

`[logic_flag]:` Format: Integer. Range: Should be one of the following integer codes: 3002 (VYES), 3003 (VNO), or 3045 (VNONE). The ***logic_flag*** is used to establish the Boolean relationship between successive expectations. If the ***logic_flag*** is set to 3002 (VYES), the current expectation and the next expectation will be connected with "OR" gate logic to form one expectation unit. Expectations connected with the logic_flag set to 3002 (VYES) count as a single expectation unit for the purposes of setting the ***number_of_expectations*** variable. When the VYES option is selected, satisfying any one of the associated expectations will satisfy the entire expectation unit. When the logic_flag is set to 3003 (VNO) or 3045 (VNONE), the current expectation and the next expectation are treated as separate expectation units and are connected by "AND" gate logic. In this case, each expectation must be satisfied in order to satisfy the complete substep expectation set. There is no limit on the number of individual expectations that can be connected with "OR" and "AND" gate logic. Thus, it is possible to create complex expectation requirements.

`nonresponse_action_type:` Format: Integer. Range: Must be one of the following integer codes:

Integer Code	Description	Comments
3045	VNONE	No contingency action is performed. If the VNONE option is used, no other data fields shall be provided for the remainder of the substep input.
3074	VACTION	Simple controller manipulation. The ***nonresponse_action_name*** controller will be positioned to the value specified by ***nonresponse_control_value***
3110	VPROCEDURE	The non-response action will transfer to the procedure step specified by the ***procedure_name*** and ***step_name*** data fields.

Integer Code	Description	Comments
3112	VADDITIVE	The ***nonresponse_action_name*** controller control value will be incremented by the amount specified by the ***nonresponse_control_value*** variable. The control value is bounded by the values for fully open/on and fully closed/off provided for the action_name control in the "ControlPanel.txt" input file.
3116	VSTOP	Terminates the accident sequence if the non-response contingency action is activated. If the VSTOP option is used, no other data fields shall be provided for the remainder of the substep input.
3139	VPARAMETER_CONTROL	The ***nonresponse_action_name*** controller will be positioned to the perceived value of the ***nonresponse_control_parameter*** indicator. If the ***nonresponse_control_parameter*** value has not been perceived by the operator, no control manipulation is performed.
3140	VPARAMETER_SCAN	The contingency will add the control panel indicator specified by the ***nonresponse_action_name*** to the operator's scan queue list.

The non-response contingency action is only initiated if the substep expectations are not met (i.e., if any expectation unit is not satisfied). This data field must be provided.

[procedure_name]: Format: String. Range: Must be associated with a valid ***procedure_name*** and ***step_name*** listed in the "Procedures.txt" input file. Spaces must not be used (use the underbar character ("_") rather than a space when needed). Should only be entered if the ***nonresponse_action_type*** is integer code 3110 (VPROCEDURE) for a procedure transfer.

[step_name]: Format: String. Range: Must be associated with a valid ***procedure_name*** and ***step_name*** listed in the "Procedures.txt" input file. Spaces must not be used (use the underbar character ("_") rather than a space when needed). Should only be entered if the ***nonresponse_action_type*** is integer code 3110 (VPROCEDURE) for a procedure transfer.

[nonresponse_action_name]: Format: Sting. Range: Varies depending on the ***nonresponse_action_type*** - if the ***nonresponse_action_type*** is VADDITIVE (integer code 3112), VACTION (integer code 3074), or VPARAMETER_CONTORL (integer code 3139), the ***nonresponse_action_name*** specifies the control to be manipulated and must refer to a valid controller listed in the "ControlPanel.txt" input file. If the ***nonresponse_action_name*** will initiate a maneuvering action, the ***nonresponse_action_name*** must include the prefix "MANEUVER_" and should reference a valid maneuvering action listed in the "KB_OAT(ODM)_Maneuvering_Actions.txt" input file. If the ***nonresponse_action_type*** is VPARAMETER_SCAN (integer code 3140), the ***nonresponse_action_name*** specifies the control panel indicator to be added to the operator's scan queue and must refer to a valid parameter, component, or alarm listed in the "ControlPanel.txt" input file. This data field is not entered if the ***nonresponse_action_type*** is VPROCEDURE (integer code 3110), VSTOP (integer code 3116), or VNONE (integer code 3045).

[nonresponse_indicator_type]: Format: Integer. Range: 1, 2, or 3. Specifies the type of parameter represented by the ***nonresponse_action_name*** variable. The operator scan queue contains three sub-queue lists: control panel parameters, components, and alarms. The ***nonresponse_indicator_type*** is used to place the ***nonresponse_action_name*** in the correct scan queue sublist. This field is only applicable when the ***nonresponse_action_type*** is set to VPARAMETER_SCAN (integer code 3140). The following options are available:

nonresponse_indicator_type **Value**	**Description**	**Comments**
1	VALARM_STATE	Control panel alarm
2	VCOMPONENT_STATE	Control panel component status indicator
3	VPARAMETER_VALUE	Control panel parameter value indicator

[nonresponse_scan_priority]: Format: Integer. Range: ≥ 1. Specifies the initial priority level applied to the ***nonresponse_action_name*** added to the operator scan queue. A lower ***nonresponse_scan_priority*** value designates a higher priority control panel indicator (i.e., the highest priority items are designated with a priority level of 1). Because the priority level can decay over time, specifying a lower ***nonresponse_scan_priority*** value will increase the residence time of the alarm on the scan queue list. This field is only applicable when the ***nonresponse_action_type*** is set to VPARAMETER_SCAN (integer code 3140)

[nonresponse_control_value]: Format: Double. Range: any, but should be consistent with the control range of the controller specified by the ***nonresponse_action_name*** variable. This data filed must be provided when the ***nonresponse_action_type*** is 3074 (VACTION) or 3112 (VADDITIVE).

[nonresponse_control_parameter]: Format: String. Range: Must match a valid ***parameter_name***, listed in the "ControlPanel.txt" input file. Specifies the control panel indicator value that will be used to provide the control value for the ***nonresponse_action_name*** controller. This field shall only be included when the ***nonresponse_action_type*** is 3139 (VPARAMETER_CONTROL).

[skip_nonresponse_alpha]: Format: Double. Range: > 0.0. Specifies the α parameter in the procedure non-response step skipping probability distribution. To capture the uncertainty associated with skipping procedure steps, the Beta Distribution is used to model the base non-response step skipping probability (Equation 16):

$$p(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{(\alpha-1)} (1-x)^{(\beta-1)}$$
$$\overline{p} = \frac{\alpha}{\alpha+\beta} \quad (\textit{mean value}) \qquad \text{(Equation 16)}$$

Two parameters must be specified, the alpha parameter (α) and the beta parameter (β). During a simulation run, the non-response step skipping probability is determined by a Monte Carlo sample of the Beta distribution. To improve the reproducibility of simulation results, it is recommended that the α and β parameters be selected to minimize the variance of the failure distribution. For example, the failure distributions p(x, α = 1. β = 1.) and p(x, α = 100. β = 100.) both have the same mean value (0.5), but the variance for the latter case is more than an order of magnitude lower. A smaller variance will yield more reproducible simulation results. As described in the "Procedures.txt" input file description, the base step skipping probability is modified by several multiplicative factors to account for status and dynamic effects. This parameter must be provided when the ***non_response_action_type*** is 3074 (VACTION) , 3110 (VPROCEDURE), 3112 (VADDITIVE), 3139 (VPARAMETER_CONTROL), or 3140 (VPARAMETER_SCAN).

[skip_nonresponse_beta] Format: Double. Range: > 0.0. Specifies the β parameter in the base non-response step skipping probability distribution. As described in the "Procedures.txt" input file description, the base step skipping probability is modified by several multiplicative factors to account for status and dynamic effects. This parameter must be provided when the ***non_response_action_type*** is 3074 (VACTION), 3110 (VPROCEDURE), 3112 (VADDITIVE), 3139 (VPARAMETER_CONTROL), or 3140 (VPARAMETER_SCAN).

next_procedure_name: Format: String. Range: Must be associated with a valid ***procedure_name*** and ***step_name*** listed in the "Procedures.txt" input file. Spaces must not be used (use the underbar character ("_") rather than a space when needed). Enter the keyword ***NONE*** if no additional procedure steps will be performed.

next_step_name: Format: String. Range: Must be associated with a valid ***procedure_name*** and ***step_name*** listed in the "Procedures.txt" input file. Spaces must not be used (use the underbar character ("_") rather than a space when needed). Enter the keyword ***NONE*** if no additional procedure steps will be performed.

4. <u>Sample Input</u>

```
E_0   Step_4.1    "Check Safety Injection Status"
3153  3165  1.0   1.0
1
A_Safety_Injection      3109  0.0   1.0   1.0   1.0   1.0   100.0
4
A_Lo_SG_Pressure_SI           3045  0.0   1.0   1.0   3023  3002
Block_Main_Steam_Isolation    3045  0.0   0.5   2.0   3022  3003
A_Hi_Cont_Pressure            3045  0.0   1.0   1.0   3023  3003
A_Lo_PZR_Pressure_SI          3045  0.0   1.0   1.0   3023  3002
Block_Low_Press_SI            3045  0.0   0.5   2.0   3022  3003
Safety_Injection              3002  0.0   1.0   1.0   3023  3045
3110
E_0   Step_4.2          1.0   100.0
ES_0.1      Step_1.1
```

The above sample input provides the proceduralized actions for Procedure E-0, Step 4.1, Check Safety Injection Status." The associated file name for this input file is "ZProcedure_E_0_Step_4.1.txt". The procedure is an emergency operating procedure (3153 - VEOP) with a step objective of diagnosis (3165 – VDIAGNOSIS). The step complexity factor is set to the neutral value of 1.0. The step consists of a single substep. The substep action verifies the status of the safety injection alarm (A_Safety_Injection) with the required time modeled by Weibull parameters $u = 0.0$, $\alpha = 1.0$, and $\beta = 1.0$. The base step skip probability is set to a mean value of ~0.01 with Beta distribution parameters $\alpha = 1.0$, $\beta = 100.0$. The substep includes of four expectation units:

- Verify that low steam generator pressure safety injection alarm is OFF **or** that the main steam isolation has been manually blocked, **and**
- Verify that the high containment pressure alarm is OFF, **and**
- Verify that the low pressurizer pressure safety injection alarm is OFF **or** that the low pressure safety injection has been manually blocked, **and**

- Verify that the safety injection alarm is OFF

If any of these expectations is not satisfied, the non-response action is initiated to transfer to Step 4.2 of Procedure E-0. If the expectations are satisfied, the operators transfer to Procedure ES-0.1, Step 1.1.

General Description:

The plant hardware related input files are used to establish the linkage between the RELAP thermal hydraulic model and the ADS-IDAC environment and manage hardware failure events. The "RELAP5_channels.txt" is used to set up communication channels between the RELAP program and the ADS-IDAC environment. These channels allow information exchanges between ADS-IDAC and RELAP. The "ControlPanel.txt" input file describes the ADS-IDAC control panel elements. Similar to an actual control room, the ADS-IDAC control panel is the main information interface between the simulated operators and the reactor plant model. ADS-IDAC allows the analyst to model two types of hardware failure events: (1) time dependent failures, and (2) conditional failures. Time dependent failures are described in the "Initiating_Event.txt" input file and allow the analyst to initiate hardware failures at a prescribed time during the simulation. Conditional failures are triggered when a specified component changes its operating state. Time dependent failures generate only a single failure event sequence branch while conditional failures generate two event sequence branches – a success path and a failure path. Both failure types permit the operators to attempt to recover the failed equipment. If a recovery is attempted, two additional branches are generated – a successful recovery branch and a permanent failure branch.

Input Files:

ControlPanel.txt
Initiating_Event.txt
RELAP5_channels.txt
SystemReliability.txt

1. <u>Purpose</u>

The "ControlPanel.txt" input file, in conjunction with the "RELAP5_channels.txt" file, describes the main control panel interface between the RELAP thermal hydraulic reactor plant model and the operator behavior model. All plant status information perceived by the operations crew must be displayed on the ADS-IDAC control panel and all control manipulations must be performed through the control panel interface.

Three main categories of information can be displayed on the ADS-IDAC control panel: (1) reactor plant thermal hydraulic parameters (e.g., temperature, pressure, flow rate), (2) component operating state (e.g., on, off, open, closed), and (3) alarms. Indicators for thermal hydraulic parameters can display both the value of an indicator or the rate of change of the target parameter. The rate of change of a parameter can be used to provide a trend display for use by the simulated operators (similar to a strip chart recorder). Component operating state information can be used to model simple panel status lights (e.g., pump operating status). Finally, alarms based on parameter values, component operating state, or the difference between two parameters can be displayed on the control panel.

Operators may manipulate two types of controllers through the ADS-IDAC control panel: (1) controls with fine adjustment capability (control values can be assigned over a range of acceptable values), and (2) simple controllers that utilize a discrete control value (e.g., open, close, off, or on). Fine adjustment controls should be used for components that can be operated over a continuous range of input values such as throttle valves or setpoint controllers. Simple discrete controllers should be used for components with binary operating states such as pumps or control switches.

Although all control panel indicators and controls can be assigned a failure value (i.e., the displayed value or state when the instrument, alarm, or control has failed), this feature has not been fully implemented in the current version of ADS-IDAC. It is recommended that the analyst use the "KB_OAT(ODM)_Bias_Factors.txt" input files to model failed control panel indicators.

The ADS-IDAC control panel provides a central link between most of the program modules within ADS-IDAC. Therefore, several other input files cross reference indicators, alarms, and controls contained in this input file. The analyst should ensure that indicators, controls, and alarms referenced by other input files use the same item name used in the ControlPanel.txt file or errors may occur during program execution.

2. Input File Format

```
Control_Panel_Parameters   number_of_parameters
Number_of_Rate_Data_Points number_of_rate_data_points
parameter_name_1    channel      indicator_type   required_time
       failure_value
.                   .            .                .                .
parameter_name_n    channel      indicator_type   required_time
       failure_value


Indicator_Heat_Structures  number_of_heat_structure_parameters
heat_structure_name_1   heat_channel     indicator_type   required_time
failure_value           heat_mesh_point
.                       .                .                .
heat_structure_name_n   heat_channel     indicator_type   required_time
failure_value           heat_mesh_point


Control_Panel_Components   number_of_components
component_name_1   trip_channel   indicator_type  required_time
failure_state
threshold_type     threshold_trip_value
.                  .
component_name_n   trip_channel   indicator_type  required_time
failure_state
threshold_type     threshold_trip_value


Control_Panel_Fine_Adjust number_of_fine_adjust_controls
control_name_1      interactive_channel       required_time
open/on_value   closed/off_value   neutral_value   control_failure_value
.                   .                   .                   .
control_name_n      interactive_channel       required_time
open/on_value   closed/off_value   neutral_value   control_failure_value


Control_Panel_Controller   number_of_controllers
control_name_1      interactive_channel       required_time
open/on_state   closed/off_state   neutral_state   control_failure_state
.                   .                   .                   .
control_name_n      interactive_channel       required_time
open/on_state   closed/off_state   neutral_state   control_failure_state


Alarm_for_Parameter_State  number_of_parameter_alarms
alarm_name_1        alarm_parameter     alarm_weight
required_time       logic               setpoint
.                   .                   .
alarm_name_n        alarm_parameter     alarm_weight
required_time       logic               setpoint


Alarm_for_Component_State  number_of_component_alarms
alarm_name_1        alarm_component     alarm_weight
required_time       state_setpoint
.                   .                   .
alarm_name_n        alarm_component     alarm_weight
required_time       state_setpoint
```

```
Alarm_Parameter_Difference number_of_difference_alarms
alarm_name_1        first_parameter     alarm_weight        required_time
                    second_parameter    diff_setpoint
.                   .                   .                   .
alarm_name_n        first_parameter     alarm_weight        required_time
                    second_parameter    diff_setpoint
```

3. Input Description

number_of_parameters: Format: Integer. Range: ≥ 0. Specifies the number of control panel indicators based on parameter values that are provided in "ControlPanel.txt" input file. Indicators based on heat structures are not included in the ***number_of_parameters*** (heat structure indicators are listed in a separate data field). The ***number_of_parameters*** must exactly match the number of listed parameter indicators or an error will occur during input file processing or program execution. Parameter based indicators are used to provide indication of basic thermal hydraulic properties on the ADS-IDAC control panel (e.g., pressure, temperature, flow, levels, etc.).

number_of_rate_data_points: Format: Integer. Range: ≥ 1. Specifies the number of parameter data points that will be used to calculate parameter trends. During each ADS-IDAC time step, the elapsed simulation time and the current parameter value for each trended parameter is added to a storage queue. Parameter trends are determined by the slope of a linear regression fit to the stored data points. Increasing the number_of_rate_data_points variable will result in more data points (over a longer time period) being used to calculate trend information. Although a larger number of data points can result in more stable trend information, it will take a longer time for an emergent trend to become evident. For the default ADS-IDAC time step of 0.5 seconds, 120 data points will provide trend information over the preceding minute of simulation time (i.e., 120 data points x 0.5 seconds/data point = 60.0 seconds). Any control panel parameter indicator can be used to provide trend information by adding the prefix "RATE_" to the ***parameter_name***.

parameter_name: Format: String. Range: any. Descriptive name for the control panel parameter indicator. Spaces must not be used (use the underbar character ("_") rather than a space when needed.

parameter_channel Format: String. Range: Must refer to a valid hydraulic volume, hydraulic junction, or control variable channel in the "RELAP5_channels.txt" input file. This parameter establishes the linkage between the ADS-IDAC control panel and the RELAP thermal hydraulic model.

indicator_type: Format: String. Range: Must refer to a valid indicator type for the selected ***parameter_channel***. The string variable should not be enclosed in quotation marks. The ***indicator_type*** variable must be supplied for the parameter, heat structure, and component state data fields. The following table lists valid indicator types:

RELAP Channel Type (***parameter_channel***, ***heat_channel***, **or** ***trip_channel***)	**Valid Indicator Types** (***indicator_type***)	**Comments**
Hydraulic Volume (***HV_xxx***)	Pressure Liquid_Temperature Vapor_Temperature Saturation_Temperature Boron_Denisty Equilibrium_Quality Flow_Regime_Number	Measurement units are determined by the value of the ***R5PAR_mp_SI_UNIT*** variable (true = SI units, false = British). See General Notes in Section 2 for more information.
Hydraulic Junction (***HJ_xxx***)	Mass_Flow_Rate Liquid_Velocity Vapor_Velocity	Measurement units are determined by the value of the ***R5PAR_mp_SI_UNIT*** variable (true = SI units, false = British). See General Notes in Section 2 for more information.
Control Variable (***CV_xxx***)	Any valid string	Units of indicator are set by RELAP control variable. The indicator type field is not used, but a dummy value must be entered. To improve readability of the input file, it is recommended that the string "Value" be used as the indicator type.
Heat Structure (***HS_xxxnn***)	Any valid string	The indicator type field is not used, but a dummy value must be entered. To improve readability of the input file, it is recommended that the string "Temperature" be used as the indicator type. Only temperature may be obtained from a heat structure channel. Measurement units are determined by the value of the ***R5PAR_mp_SI_UNIT*** variable (true = SI units, false = British). See General Notes in Section 2 for more information.
Component State (***VT_xxx or LT_xxx***)	Any valid string	The indicator type field is not used, but a dummy value must be entered. To improve readability of the input file, it is recommended that the string "Trip_Time" be used as the indicator type. The component state RELAP channel returns the time that the associated trip was first set to true (or -1 if the trip is false).

`required_time:` Format: Double. Range: ≥ 0.0. Originally intended to specify the time required for the operator to read a control panel indicator, check an alarm status, or operate a controller. The ***required_time*** variable is not used since the time required to perform these activities is now set in the procedure step input files ("ZProcedure_" series). Although this feature is not used in the current version of ADS-IDAC, the analyst must provide a dummy input value. A

required_time variable must be provided for parameter, heat structure, component, controls, and alarm data fields.

failure_value: Format: Double. Range: any. Originally intended to specify the value indicated by the associated indicator or fine control after a failure. The ***failure_value*** variable is not used for indicators since an instrument failure state can now be set by an appropriate bias factor in the operator perception filter ("KB_OAT(ODM)_Bias_Factors.txt" input file). Although this feature is not used to model indicator failures in the current version of ADS-IDAC, the analyst must provide a dummy input value. A ***failure_value*** variable must be provided for parameter and heat structure data fields.

number_of_heat_structure_parameters: Format: Integer. Range: ≥ 0. Specifies the number of control panel indicators based on heat structure values that are provided in "ControlPanel.txt" input file. The ***number_of_heat_structure_parameters*** must exactly match the number of listed heat structure indicators or an error will occur during input file processing or program execution. Heat structure indicators are used to indicate the temperature of heat sources or sinks included in the RELAP thermal hydraulic model (e.g., fuel or clad temperature, steam generator shell temperature, or reactor vessel temperature).

heat_structure_name: Format: String. Range: any. Descriptive name for the control panel heat structure indicator. Spaces must not be used (use the underbar character ("_") rather than a space when needed.

heat_channel: Format: String. Range: Must refer to a valid heat structure channel in the "RELAP5_channels.txt" input file. This parameter establishes the linkage between the ADS-IDAC control panel and the RELAP thermal hydraulic model.

heat_mesh_point: Format: Integer. Range: ≥ 0. Specifies the heat structure mesh point within the heat structure channel. Must refer to a valid heat structure mesh point number or an error will occur during input file processing.

number_of_components: Format: Integer. Range: ≥ 0. Specifies the number of control panel component status indicators provided in "ControlPanel.txt" input file. The ***number_of_components*** must exactly match the number of listed component state indicators or an error will occur during input file processing or program execution. Component state indicators are used to indicate the state of components or control systems (e.g., pump running, valve closed, reactor tripped, etc). Because component state indicators are based derived from Boolean variables, they can only indicate two possible state values.

component_name: Format: String. Range: any. Descriptive name for the control panel component state indicator. Spaces must not be used (use the underbar character ("_") rather than a space when needed.

`trip_channel`: Format: String. Range: Must refer to a variable or logical trip channel in the "RELAP5_channels.txt" input file. This parameter establishes the linkage between the ADS-IDAC control panel and the RELAP thermal hydraulic model.

`failure_state:` Format: String. Range: Only the following strings are valid – ON, OFF, OPEN, CLOSE, and TRIP. Originally intended to specify the status indicated by the associated component after a failure. The ***failure_state*** variable has not been fully implemented in ADS-IDAC. Although this feature is not used in the current version of ADS-IDAC, the analyst must provide a dummy input value.

`threshold_type` Format: String. Range: Only the following strings constitute valid ***threshold_type*** values:

GREATER_THEN_ON
SMALLER_THEN_ON
GREATER_THEN_OPEN
SMALLER_THEN_OPEN
GREATER_THEN_OFF
SMALLER_THEN_OFF
GREATER_THEN_CLOSE
SMALLER_THEN_CLOSE

The ***threshold_type*** variable was originally intended to accommodate binary state devices such as pumps or trip valves. Thus the component state indicator values for elements with ***threshold_type*** with an "_ON" or "_OFF" suffix are VON or VOFF (integer codes 3022 or 3023). Component state indicator values for elements with ***threshold_type*** with an "_OPEN" or "_CLOSE" suffix are VOPEN or VCLOSE (integer codes 3098 or 3099).

The RELAP variable and logical trip channels can take two possible types of values: (1) if the trip channel is TRUE, the trip channel value is the time that the associated trip was last set to a TRUE state, and (2) if the trip channel is FALSE, the trip channel value is set to -1. In order to accommodate different interpretations of the meaning of a TRUE or FALSE trip channel value, ADS-IDAC allows the analyst to choose an appropriate ***threshold_type*** variable. For example, the TRUE state of a variable or logical trip channel could represent a component state of ON or OFF, depending on the underlying logic used in the RELAP input deck. There are only four unique threshold_type variable categories, but the analyst may find it easier using one of the companion options (e.g., SMALLER_THEN_OFF vice GREATER_THEN_ON) depending on the default state of the associated component. In addition, the analyst can delay the change in the state indicated on the control panel when a component changes its actual operating state. The following table provides a detailed summary of the available ***threshold_type*** options:

Threshold Type Categories	Component State Required to Set Trip Channel to TRUE in RELAP Input Deck	Component State when RELAP Trip Channel is False	Component State when RELAP Trip Channel is True
GREATER_THEN_ON or SMALLER_THEN_OFF	ON	OFF	**ON** when elapsed time since the trip last set to TRUE is greater than the threshold value
GREATER_THEN_OFF or SMALLER_THEN_ON	OFF	ON	**OFF** when elapsed time since the trip last set to TRUE is greater then the threshold value
GREATER_THEN_OPEN or SMALLER_THEN_CLOSE	OPEN	CLOSE	**OPEN** when elapsed time since the trip last set to TRUE is greater than the threshold value
GREATER_THEN_CLOSE or SMALLER_THEN_OPEN	CLOSE	OPEN	**CLOSE** when elapsed time since the trip last set to TRUE is greater then the threshold value

For most problems, it is recommended that the analyst use either the "GREATER_THEN_ON" or "GREATER_THEN_OPEN" ***threshold_type*** in order to increase the readability of the input file. In certain cases, it may be necessary to modify the underlying trip logic in the RELAP input deck to accommodate these ***threshold_type*** options.

threshold_trip_value: Format: Double. Range: ≥ 0.0. Specifies the time delay between an actual change in component operating state and the indication of the state change on the control panel.

number_of_fine_adjust_controls: Format: Integer. Range: ≥ 0. Specifies the number of control panel control elements with fine adjustment capability listed in "ControlPanel.txt" input file. The ***number_of_ fine_adjust_controls*** must exactly match the number of listed fine adjustment controls or an error will occur during input file processing or program execution. Controllers with fine adjustment capability are used to model components with a continuous range of operating states such as throttle valves and setpoint controllers.

control_name: Format: String. Range: any. Descriptive name for the control panel control element. Spaces must not be used (use the underbar character ("_") rather than a space when needed. The ***control_name*** variable must be supplied for control panel fine adjustment controls and binary state controllers.

interactive_channel: Format: String. Range: Must refer to a valid interactive control channel in the "RELAP5_channels.txt" input file. This parameter establishes the linkage between the ADS-IDAC control panel and the RELAP thermal hydraulic model.

open/on_value: Format: Double. Range: any. Specifies the upper control limit for the fine adjustment controller.

`closed/off_value:` Format: Double. Range: any. Specifies the lower control limit for the fine adjustment controller.

`neutral_value:` Format: Double. Range: any. Specifies the neutral control setting for the fine adjustment controller.

`control_failure_value:` Format: Double. Range: any. Specifies the failure control value setting for the fine adjustment controller. If a failure of the associated fine adjustment control is activated by a component status change referenced in the "SystemReliability.txt" input file, the controller is set to the ***control_failure_value***.

`number_of_controllers:` Format: Integer. Range: ≥ 0. Specifies the number of control panel binary state controllers listed in the "ControlPanel.txt" input file. The ***number_of_controllers*** must exactly match the number of listed binary state controllers or an error will occur during input file processing or program execution. Binary state controllers are used to operate two state devices such as pumps and certain control switches.

`open/on_state:` Format: Integer. Range: any. Specifies the open or on state for the binary state controller.

`closed/off_state:` Format: Integer. Range: any. Specifies the closed or off state for the binary state controller.

`neutral_state:` Format: Integer. Range: any. Specifies the neutral control setting for the fine adjustment controller.

`control_failure_state:` Format: String. Range: Only the following strings constitute valid control failure states – ON, OFF, OPEN, CLOSE, and NONE. Specifies the failure control state setting for the binary state controller. If a failure of the associated binary state control is activated by a component status change referenced in the "SystemReliability.txt" input file, the controller is set to the control state associated with the ***control_failure_state***. For example, if the ***control_failure_state*** was set to "ON" and the ***open/on_state*** was 1, the binary state control value would be set to 1 upon a failure activation. The following table summarizes controller failure states:

control_failure_state	Failure Control Value
OPEN or ON	`open/on_state`
CLOSE or OFF	`closed/off_state`
NONE	`neutral_state`

`number_of_parameter_alarms:` Format: Integer. Range: ≥ 0. Specifies the number of parameter-based alarms provided in "ControlPanel.txt" input file. The ***number_of_ parameter_alarms*** must exactly match the number of listed parameter-based alarms or an error will occur during input file processing or program execution. Parameter alarms are used to alert operators when thermal hydraulic parameters exceed a preset threshold (e.g., low reactor pressure, high reactor power, etc).

`alarm_name:` Format: String. Range: any. Descriptive name for the control panel parameter alarm. Spaces must not be used (use the underbar character ("_") rather than a space when needed. It is recommended that the prefix "A_" be used to distinguish alarms from other control panel indicators. Using the prefix "A_ENDSEQ" will cause the sequence to terminate upon alarm activation. The ***alarm_name*** variable must be included with parameter, component, and difference alarms.

`alarm_parameter:` Format: String. Range: The ***alarm_parameter*** must match a control panel parameter or heat structure indicator listed in the "ControlPanel.txt" input file. Specifies the parameter monitored by the alarm.

`alarm_weight:` Format: Double. Range: any. Specifies the weighting importance factor for the alarm. This feature has not been fully implemented in ADS-IDAC. The analyst should enter a suitable dummy value for this parameter. The ***alarm_weight*** variable must be included with parameter, component, and difference alarms.

`logic:` Format: String. Range: the ***logic*** variable must be one of the following or an error will occur during input file processing –

GT – greater than
GE – greater than or equal to
EQ – equal to
LE – less than or equal to
LT – less than

Quotation marks should not be used. The alarm_parameter value is compared to the setpoint using the logic factor and alarm is activated when the relationship is true. For example, if the ***logic*** variable is set to **GE**, the alarm is activated if the ***alarm_parameter*** value is equal to or greater than the ***setpoint***.

`setpoint:` Format: Double. Range: any. Specifies the parameter-based alarm setpoint.

`number_of_component_alarms:` Format: Integer. Range: ≥ 0. Specifies the number of component status alarms listed in the "ControlPanel.txt" input file. The ***number_of_component_alarms*** must exactly match the number of listed component state alarms or an error will occur during input file processing or program execution. Component state alarms are used to alert the operators to changes in component or system operating status (e.g., turbine tripped, reactor tripped, etc).

`alarm_component:` Format: String. Range: The ***alarm_component*** must match a control panel component status indicator listed in the "ControlPanel.txt" input file. Specifies the component status monitored by the alarm.

`state_setpoint:` Format: String. Range: the ***state_setpoint*** variable must be set to either ON, OFF, OPEN, or CLOSE or an error will be generated during input file processing. Quotation marks should not be used. When the state of the associated ***alarm_component*** is equal to the ***state_setpoint***, the component-based alarm will be activated.

`number_of_difference_alarms:` Format: Integer. Range: ≥ 0. Specifies the number of control panel parameter difference alarms listed in the "ControlPanel.txt" input file. The ***number_of_ difference_alarms*** must exactly match the number of listed difference alarms or an error will occur during input file processing or program execution. Parameter difference alarms are used to alert the operators when two parameters, which should normally be equivalent, diverge. Example of difference alarms include main steam flow/main feed flow mismatch alarm, or pressurizer level deviation alarm (actuated when the difference between the actual pressurizer level deviates from the level control setpoint).

`first_parameter:` Format: String. Range: The ***first_parameter*** must match a control panel parameter or heat structure indicator listed in the "ControlPanel.txt" input file. Specifies the first input parameter monitored by the alarm.

`second_parameter:` Format: String. Range: The ***second_parameter*** must match a control panel parameter or heat structure indicator listed in the "ControlPanel.txt" input file. Specifies the second input parameter monitored by the alarm.

`difference_setpoint:` Format: Double. Range: any. Specifies the parameter difference-based alarm setpoint. The parameter difference alarm is activated when the difference between the first and second parameters exceeds the difference setpoint:

```
first_parameter – second_parameter > difference_setpoint
```

The parameter difference alarm is only activated by a one-sided deviation (i.e., the alarm will not activate if the ***first_parameter*** is less than the ***second_parameter***, regardless of the magnitude of the difference). If it is necessary to detect two-sided deviations, two parameter difference alarms should be set up – one to detect when the first_parameter exceeds the second_parameter and another with the first and second parameters reversed (which will detect when the ***second_parameter*** is greater than the ***first_parameter***). For example, the following difference alarms will alert the operators when the A steam generator water level deviates by more than 5% (0.05) from the SG level setpoint:

```
A_SG_A_Level_Lo_Dev      SG_Level_Setpoint  0.5   20.0   SG_A_NR_Level        0.05
A_SG_A_Level_Hi_Dev      SG_A_NR_Level      0.5   20.0   SG_Level_Setpoint    0.05
```

4. Sample Input

```
Control_Panel_Parameter                  14
Number_of_Rate_Data_Points               120
```

```
Time                    CV_002   Value                 20.0 0.0
Core_Power              CV_100   Value                 20.0 0.0
SUR                     CV_491   Value                 20.0 1.0
Loop_A_Tcold            HV_216   Liquid_Temperature    20.0 0.0
Loop_A_Tave             CV_101   Value                 20.0 0.0
Loop_A_Thot             HV_204   Liquid_Temperature    20.0 0.0
PZR_Pressure            HV_340   Pressure              20.0 0.0
RATE_PZR_Pressure       HV_340   Pressure              20.0 0.0
PZR_Level               CV_202   Value                 20.0 0.0
RATE_PZR_Level          CV_202   Value                 20.0 0.0
SG_A_NR_Level           CV_502   Value                 20.0 0.0
SG_A_FW_Flow            HJ_527   Mass_Flow_Rate        20.0 0.0
SG_A_MS_Flow            HJ_282   Mass_Flow_Rate        20.0 0.0
PZR_Level_Setpoint      CV_188   Value                 20.0 0.0

Indicator_Heat_Structure_Value          1
THS_57001               HS_57001 Temperature           20.0 0.0      1

Control_Panel_Component_State           3
Reactor_Trip        LT_1698  Trip_Time   20.0   OFF  GREATER_THEN_ON
1.0E-5
Safety_Injection    LT_1669  Trip_Time   20.0   ON   GREATER_THEN_ON
1.0E-5
Turbine_Trip        LT_1602  Trip_Time   20.0   ON   GREATER_THEN_ON
1.0E-5

Controls_Panel_Fine_Adjust                    3
X_PZR_Spray_Valve         IC_804   1.0   1.0   0.0  -1.0   0.0
X_PZR_PORV                IC_805   1.0   0.33  0.0  -1.0   0.0
X_SG_A_Atmos_PORV         IC_806   1.0   1.0   0.0  -1.0   0.0

Control_Panel_Controller                      2
X_RCP_A                   IC_821   1.0   1    -1    -1   ON
X_ACC_A_Outlet_Valve      IC_850   1.0   1     0    -1   ON

Alarm_for_Parameter_State               5
A_SG_A_Lo_Level           SG_A_NR_Level    0.5 20.0 LT     0.25
A_SG_A_LoLo_Level         SG_A_NR_Level    0.5 20.0 LT     0.12
A_SG_A_Hi_Level           SG_A_NR_Level    0.5 20.0 GT     0.5
A_PZR_Lo_Pressure         PZR_Pressure     0.5 20.0 LT     2100.0
A_ENDSEQ_Parameter        Core_Power       0.5 20.0 LT     0.001


Alarm_for_Component_State               3
A_Reactor_Trip                 Reactor_Trip           0.5    1.0   ON
A_Safety_Injection             Safety_Injection       0.5   20.0   ON
A_Turbine_Trip                 Turbine_Trip           0.5   20.0   ON

Alarm_for_Difference_Between_two_Values       2
A_SG_A_MF_Flow_Lo   SG_A_MS_Flow   0.5   20.0   SG_A_FW_Flow
350.0
A_PZR_Level_Hi_Dev  PZR_Level      0.5   20.0   PZR_Level_Setpoint
0.05
```

This sample input provides examples of each type of control panel element. In addition, each referenced RELAP channel must refer to a valid entry in the “RELALP5_channels.txt” input file.

1. Purpose

As the file name suggests, the Initiating_Event.txt input file is used to specify the event initiators to be included in the simulation. Initiating events are activated based on the elapsed simulation time and generate only a single component failure branch (i.e., no success branch is generated). As such, the sum of all sequence probabilities generated by a simulation can be no greater than the largest initiating event probability included in the simulation. The operators may attempt to recover an initiating event failure one time – in this case branches for a successful equipment recovery and a permanent equipment failure are generated.

Initiating events are distinguished from conditional hardware failures identified in the SystemReliabiltiy.txt input file by the following characteristics:

Characteristic	Initiating Event	Conditional Failure
Input File	Initiating_Event.txt	SystemReliabiltiy.txt
Activation	Simulation Time	Component State
Number of Initial Branches Generated	One (failure only)	Two (success and failure branches)
Recovery Possible?	Yes	Yes

Initiating events should be used to model time-dependent failures. A conditional failure should be used when it is necessary to model a demand failure or a failure initiated by a change in system state.

2. Input File Format

```
Initiating_Event   number_of_initiating_events
event_name_1       control_value      time        control_name
alpha_init         beta_init          alpha_rec   beta_rec
.                  .                  .           .
.                  .                  .           .
event_name_n       control_value      time        control_name
alpha_init         beta_init          alpha_rec   beta_rec
```

3. Input Description

number_of_initiating_events: Format: Integer. Range: ≥ 0. Specifies the number of initiating events provided in the input file. Initiating events are read by the program in the order that they are listed in the input file. Therefore, if more than the specified ***number_of_initiating_events*** are described in the

“InitiatingEvent.txt” input file, only the first ***number_of_initiating_events*** are read.

Changing the ***number_of_initiating_events*** variable provides a convenient method for enabling and disabling initiating events. The analyst can disable all initiating events by setting the ***number_of_initiating_events*** variable to 0 (it is not necessary to delete each initiating event from the file). If only one initiating event will be activated, the desired event is simply moved to the beginning of the initiating event list and the ***number_of_initiating_events*** variable is set to 1 (all other initiating events will be ignored and need not be removed from the input file).

event_name Format: String. Range: any. Descriptive name for the initiating event. Spaces must not be used (use the underbar character (“_”) rather than a space when needed.

control_value: Format: Double. Range: any. Specifies the control value that will be applied to the controller identified the ***control_name*** when the initiating event is activated

time: Format: Double. Range: any. This parameter specifies simulation time that the initiating event will be activated. Due to variations in the RELAP time step, the initiating event may not be activated exactly at the specified ***time*** value, but instead will be activated at the first time step with an elapsed simulation time greater than the ***time*** variable.

If a time less than 0.0 is specified, the initiating event will not be activated during the simulation. This provides the analyst with a convenient method for disabling initiating events without deleting them from the input file.

control_name: Format: String. Range: The ***control_name*** must match a control panel controller listed in the “ControlPanel.txt” input file. Specifies the controller associated with the initiating event. Spaces must not be used in the ***control_name*** (the underbar character “_” may be substituted for a space when needed).

alpha_init: Format: Double. Range: > 0.0. Specifies the α parameter in the initiating event failure distribution. To capture the uncertainty associated with initiating events, the Beta Distribution is used to model the initiating event probability (Equation 17):

$$p(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{(\alpha-1)} (1-x)^{(\beta-1)}$$

$$\overline{p} = \frac{\alpha}{\alpha+\beta} \quad (\textit{mean value}) \qquad \text{(Equation 17)}$$

Two parameters must be specified, the alpha parameter (α) and the beta parameter (β). During a simulation run, the initiating event probability is determined by a Monte Carlo sample of the Beta distribution. To improve the reproducibility of simulation results, it is recommended that the α and β parameters be selected to minimize the variance of the failure distribution. For example, the failure distributions p(x, α = 1. β = 1.) and p(x, α = 100. β = 100.) both have the same mean value (0.5), but the variance for the latter case is more than an order of magnitude lower. A smaller variance will yield more reproducible simulation results.

beta_init: Format: Double. Range: > 0.0. Specifies the β parameter in the initiating event probability distribution.

alpha_rec: Format: Double. Range: > 0.0. Specifies the α parameter in the initiating event recovery distribution. After an initiating event failure event is activated, the operator may attempt to return the failed equipment to a functional status by performing manual actions. An operator may attempt to recovery a failed component only one time. If a recovery attempt is made, two sequence branches are generated: one branch for successful recovery and one branch for a permanent (unrecoverable) failure. To capture the uncertainty associated with recovery estimates, the Beta Distribution is used to model the recovery probability (Equation 18):

$$p(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{(\alpha-1)} (1-x)^{(\beta-1)}$$

$$\overline{p} = \frac{\alpha}{\alpha+\beta} \qquad (mean\ value)$$

(Equation 18)

Two parameters must be specified, the alpha parameter (α) and the beta parameter (β). During a simulation run, the recovery probability is determined by a Monte Carlo sample of the failure distribution. To improve the reproducibility of simulation results, it is recommended that the α and β parameters be selected to minimize the variance of the failure distribution. For example, the failure distributions p(x, α = 1. β = 1.) and p(x, α = 100. β = 100.) both have the same mean value (0.5), but the variance for the latter case is more than an order of magnitude lower. A smaller variance will yield more reproducible simulation results.

In some cases, a recovery from a failure event is not physically possible or realistic. For example, the operators would not be able to recover from a steam generator tube rupture by simply "turning off" the rupture flow by resetting the control value for the appropriate control. Because recovery parameters must be specified for all failure events, it is recommended that the analyst use easily recognizable dummy values for the alpha and beta parameters (e.g., 1.0) or set the recovery probability at a very low value (e.g., α = 1.0 and β = 10^6).

`beta_rec:` Format: Double. Range: > 0.0. Specifies the β parameter in the initiating event recovery distribution.

4. Sample Input

```
Initiating_Event        3
Loss_of_Offsite_Power   1.0          -25.0        X_LOOP
            1.0         100.0        6.0          4.0
SG_A_SGTR               1.0          -60.0        X_SGTR_SG_A
            99.0        1.0          1.0          1.0
SG_B_MSLB               1.0           60.0        X_SG_B_MSLB
            1.0         100.0        6.0          4.0
```

This sample input identifies three initiating events, only one of which will be actuated during the simulation. At a simulation time of 60.0 seconds, the "SG_B_MSLB" initiating event will be activated by setting the control value for the "X_SG_B_MSLB" controller to 1.0. The mean probability of this event is 0.99. Although recovery parameters are specified for this event, these are simply dummy values and do not represent a true failure recovery probability. Because the "Loss_of_Offsite_Power" and 'SG_A_SGTR" events are have a negative event time, they will be read by the program during input file processing but not initiated.

1. Purpose

The "ControlPanel.txt" and "RELAP5_channels.txt" input files map specific control panel elements in ADS-IDAC to the RELAP thermal hydraulic model. The "RELAP5_channels.txt" input file establishes the communication channels necessary to pass information between RELAP thermal hydraulic simulation and the ADS-IDAC control panel. Seven types of communication channels can be used:

- Hydraulic Volume – Hydraulic volume channels are denoted by the prefix "HV_" and are used to establish a communication link to RELAP volumetric elements. Hydraulic volume channels allow the analyst to send volumetric quantities such as temperature, pressure, void fraction, and vapor quality to the ADS-IDAC control panel.

- Hydraulic Junctions – Hydraulic junction channels are denoted by the prefix "HJ_" and are used to establish a communication link to RELAP junction elements. Junction elements connect two or more hydraulic volume elements within a RELAP model. Hydraulic junction channels allow the analyst to send junction quantities such as mass flow rate to the ADS-IDAC control panel.

- Variable Trips – Variable trip channels are denoted by the prefix "VT_" and are used to establish a communication link to RELAP variable trips. Variable trips change state when a target parameter exceeds a preset threshold and are defined by RELAP input deck cards 401-599 (or cards 20600010-20610000 if the expanded trip option is selected). Variable trip channels are useful for setting up alarms and component state indicators on the ADS-IDAC control panel.

- Logical Trips – Logical trip channels are denoted by the prefix "LT_" and are used to establish a communication link to RELAP logical trips. Logical trips are used to combine two or more variable and logical trips to build a logical expression using Boolean operators. Logical trips are defined by RELAP input deck cards 600-799 (or cards 20610010-20620000 if the expanded trip option is selected). Variable trip channels are useful for setting up alarms and component state indicators on the ADS-IDAC control panel.

- Interactive Controls – Interactive control channels are denoted by the prefix "IC_" and are used to establish a communication link to RELAP interactive variables. Interactive variables permit data originating from an external source to be used by the RELAP code. Interactive variables are defined in cards 800 – 999 in the RELAP input deck. Interactive control channels provide the only means available for ADS-IDAC to change the value of a variable in RELAP thermal-hydraulic model. Interactive control channels allow the analyst to develop controllable components on the ADS-IDAC control panel such as pumps, valves, and actuation switches.

- Heat Structures – Heat structure channels are denoted by the prefix "HS_" and are used to establish a communication link to RELAP model heat structures. Heat structure elements are used to model heat sources and sinks such as fuel pins, reactor vessel walls, steam generator shells, or pressurizer heaters. Heat structure channels allow heat structure temperatures to be indicated on the ADS-IDAC control panel.

- Control Variable – Control variable channels are denoted by the prefix "CV_" and are used to establish a communication link to RELAP control variables. Control variables are defined by the 205 card series in the RELAP input deck and are used to perform calculations and model automatic control systems. The analyst can use control variable channels to indicate quantities derived from other basic variables on the ADS-IDAC control panel such as average reactor coolant temperature, total system flow for multiple train systems, or subcooling margin.

Each of these channels permits only one-way communication. In general, the channels provide a communication from the RELAP thermal hydraulic model to the ADS-IDAC control panel. Only the interactive control channel provides a communication link from the control panel to the thermal hydraulic model.

2. Input File Format

```
Hydraulic_Volumes              number_of_hydraulic_volumes
hydraulic_volume_name_1        hv_card_number_1
.                              .
hydraulic_volume_name_n        hv_card_number_n

Hydraulic_Junctions            number_of_hydraulic_junctions
hydraulic_junction_name_1      hj_card_number_1
.                              .
hydraulic_junction_name_n      hj_card_number_n

Variable_Trips                 number_of_variable_trips
variable_trip_name_1           vt_trip_number_1
.                              .
variable_trip_name_n           vt_trip_number_n

Logical_Trips                  number_logical_trips
logical_trip_name_1            lt_trip_number_1
.                              .
logical_trip_name_n            lt_trip_number_n

Interactive_Controls           number_of_interactive_controls
```

```
interactive_control_name_1    interactive_control_number_1
.                             .
interactive_control_name_n    interactive_control_number_n

Heat_Structures               number_of_heat_structures
heat_structure_name_1         hs_card_number_1  mesh_point_1
.                             .
heat_structure_name_n         hs_card_number_n  mesh_point_n

Control_Variables             number_of_control_variables
control_variable_name_1       control_variable_number_1
      units_1
.                             .                         .
control_variable_name_n       control_variable_number_n
      units_n
```

3. Input Description

number_of_hydraulic_volumes: Format: Integer. Range: ≥ 0. Specifies the number of hydraulic volume channels that will be described in the input file. If no hydraulic volume channels will be entered, the ***number_of_hydraulic_volumes*** should be 0 and no other hydraulic volume data should be entered in the file (i.e., do not include values for the ***hydraulic_volume_name*** or ***hv_card_number)***.

hydraulic_volume_name: Format: String. Range: any. Spaces must not be used in the ***hydraulic_volume_name*** (the underbar character "_" may be substituted for a space when needed). To improve the readability of the input files, it is recommended that the name consist of the prefix "HV_" followed by a three number derived from the RELAP card number associated with the volume. For example, the ***hydraulic_volume_name*** for the channel associated with hydraulic volume 204 (input deck card number 2040000) would be "HV_204".

hv_card_number: Format: Integer. Range: Must refer to a valid hydraulic volume in the specified RELAP input deck. The ***hv_card_number*** variable identifies the hydraulic component in the RELAP input deck associated with the channel. The card number is a nine digit integer of the form ***nnnxx0000.*** The digits ***nnn*** identifies the volume number and ***xx*** refers to the subvolume number. The volume number must refer to the first three digits of "Component Name and Type" definition card for the desired hydraulic volume. For example, if the hydraulic volume channel is associated with the volume definition card 2040000, ***nnn*** must be equal to 204. For most simple hydraulic volumes with a single subvolume, ***xx*** is "01". For hydraulic volumes comprised of more than one subvolume, ***xx*** should refer to the subvolume of interest.

number_of_hydraulic_junctions: Format: Integer. Range: ≥ 0. Specifies the number of hydraulic junction channels that will be described in the input file. If no hydraulic junction channels will be entered, the ***number_of_hydraulic_junctions*** should be 0 and no other hydraulic junction data should be entered in the file (i.e., do not include values for the ***hydraulic_junction_name*** or ***hj_card_number***).

hydraulic_junction_name: Format: String. Range: any. Spaces must not be used in the ***hydraulic_junction_name*** (the underbar character "_" may be substituted for a space when needed). To improve the readability of the input files, it is recommended that the name consist of the prefix "HJ_" followed by a three digit number derived from the RELAP card number associated with the volume. For example, the ***hydraulic_junction_name*** for the channel associated with hydraulic junction 205 (input deck card number 2050000) would be "HJ_205".

hj_card_number: Format: Integer. Range: Must refer to a valid hydraulic junction in the specified RELAP input deck. The ***hj_card_number*** variable identifies the hydraulic junction component in the RELAP input deck associated with the channel. The card number is a nine digit integer of the form ***nnnxx0000.*** The digits ***nnn*** identifies the junction number and ***xx*** refers to the subjunction number. The junction number must refer to the first three digits of "Component Name and Type" definition card for the desired hydraulic junction. For example, if the hydraulic junction channel is associated with the junction definition card 2050000, ***nnn*** must be equal to 205. For most simple hydraulic junctions, ***xx*** is "00". If the component has multiple subjunctions (such as a branch component), ***xx*** should refer to the subjunction of interest.

number_of_variable_trips: Format: Integer. Range: ≥ 0. Specifies the number of variable trip channels that will be described in the input file. If no variable trip channels will be entered, the ***number_of_variable_trips*** should be 0 and no other variable trip data should be entered in the file (i.e., do not include values for the ***variable_trip_name*** or ***vt_card_number***)

variable_trip_name: Format: String. Range: any. Spaces must not be used in the ***variable_trip_name*** (the underbar character "_" may be substituted for a space when needed). To improve the readability of the input files, it is recommended that the name consist of the prefix "VT_" followed by a three or four digit number derived from the RELAP card number associated with the variable trip. For example, the ***variable_trip_name*** for the channel associated with trip 450 (input deck card number 0000450 or 20604500 in expanded format) would be "VT_450".

vt_trip_number: Format: Integer. Range: Must refer to a valid variable trip in the specified RELAP input deck[12]. The ***vt_trip_number*** variable identifies the variable trip component in the RELAP input deck associated with the channel. The trip number is an integer in the form ***nnnn*** and is derived from the card number describing the variable trip. For example, the ***vt_trip_number*** for the variable trip described by card 0000**450** (or 2060**450**0 for expanded format)

[12] Valid variable trip numbers are 401-599 (or 1-1000 for expanded trip number format).

would be 450. It is not necessary to enter leading zeros if the trip number is less than 1000.

number_logical_trips: Format: Integer. Range: ≥ 0. Specifies the number of logical trip channels that will be described in the input file. If no logical trip channels will be entered, the ***number_of_logical_trips*** should be 0 and no other logical trip data should be entered in the file (i.e., do not include values for the ***logical_trip_name*** or ***lt_card_number***)

logical_trip_name: Format: String. Range: any. Spaces must not be used in the ***logical_trip_name*** (the underbar character "_" may be substituted for a space when needed). To improve the readability of the input files, it is recommended that the name consist of the prefix "LT_" followed by a three or four digit number derived from the RELAP card number associated with the variable trip. For example, the ***variable_trip_name*** for the channel associated with trip 450 (input deck card number 0000450 or 20604500 in expanded format) would be "VT_450".

lt_trip_number: Format: Integer. Range: Must refer to a valid logical trip in the specified RELAP input deck[13]. The ***lt_trip_number*** variable identifies the logical trip component in the RELAP input deck associated with the channel. The trip number is an integer in the form ***nnnn*** and is derived from the card number describing the logical trip. For example, the ***lt_trip_number*** for the variable trip described by card 0000**650** would be 650. If expanded trip numbering format is used, the ***lt_trip_number*** for the variable trip described by card 206**1650**0 would be 1650. It is not necessary to enter leading zeros if the trip number is less than 1000 (applicable to non-expanded trip number format only).

number_of_interactive_controls: Format: Integer. Range: ≥ 0. Specifies the number of interactive variable channels that will be described in the input file. If no interactive variable channels will be entered, the ***number_of_interactive_controls*** should be 0 and no other interactive variable data should be entered in the file (i.e., do not include values for the ***interactive_control_name*** or the ***interactive_control_number***).

interactive_control_name: Format: String. Range: any. Spaces must not be used in the ***hydraulic_junction_name*** (the underbar character "_" may be substituted for a space when needed). To improve the readability of the input files, it is recommended that the name consist of the prefix "IC_" followed by a three digit number derived from the RELAP card number associated with the interactive variable. For example, the ***interactive_control_name*** for the channel associated with interactive variable 850 (input deck card number 0000850) would be "IC_850".

interactive_control_number: Format: Integer. Range: Must refer to a valid interactive variable in the specified RELAP input deck (i.e., 801 – 999). The

[13] Valid logical trip numbers are 601-799 (or 1001 – 2000 for expanded trip number format)

interactive_control_number variable identifies the interactive component in the RELAP input deck associated with the channel. The card number is a three digit integer of the form ***nnn*** where ***nnn*** is equivalent to the interactive variable card number in the input deck. For example, the interactive***_control_number*** for the interactive variable defined by card 0000850 is simply 850.

number_of_heat_structures: Format: Integer. Range: ≥ 0. Specifies the number of heat structure channels that will be described in the input file. If no heat structure channels will be entered, the ***number_of_heat_structures*** should be 0 and no other heat structure data should be entered in the file (i.e., do not include values for the ***heat_structure_name, hs_card_number,*** or the ***mesh_point***).

heat_structure_name: Format: String. Range: any. Spaces must not be used in the ***heat_structure_name*** (the underbar character "_" may be substituted for a space when needed). To improve the readability of the input files, it is recommended that the name consist of the prefix "HS_" followed by a five digit number in the form CCCNN derived from the RELAP card number associated with the heat structure. CCC refers to the heat structure component number and NN refers to the heat structure axial node number. For example, the ***heat_structure_name*** for the channel associated with node 8 of heat structure 850 would be "HS_85008".

hs_card_number: Format: Integer. Range: Must refer to a valid heat structure in the specified RELAP input deck. The ***hs_card_number*** variable identifies the heat structure component and axial node number in the RELAP input deck associated with the channel. The card number is a seven digit integer in the form of ***cccg0nn.*** The digits ***ccc*** identify the heat structure number, ***g*** refers to the geometry number, and ***nn*** refer to the axial node number. The heat structure number and geometry number must be consistent with the "General Heat Structure" card number for the associated heat structure in the RELAP input deck. For example, if the heat structure channel is associated with axial node 5 of the heat structure defined by card 12051000, the ***hs_card_number*** is 2051005

mesh_point: Format: Integer. Range: Must refer to a valid heat structure radial mesh point in the specified RELAP input deck. Specifies the mesh point number for the heat structure channel.

number_of_control_variables: Format: Integer. Range: ≥ 0. Specifies the number of control variable channels that will be described in the input file. If no control variable channels will be entered, the ***number_of_control_variables*** should be 0 and no other control variable data should be entered in the file (i.e., do not include values for the ***control_variable_name*** or the ***control_variable_number***).

control_variable_name: Format: String. Range: any. Spaces must not be used in the ***control_variable_name*** (the underbar character "_" may be substituted for a space when needed). To improve the readability of the input files, it is recommended that the name consist of the prefix "CV_" followed by a three digit number derived from the RELAP 205 series card number associated with the control variable. For example, the ***control_variable_name*** for the channel associated with variable 535 (input deck card number 205**535**000) would be "CV_535".

control_variable_number: Format: Integer. Range: Must refer to a valid control variable in the specified RELAP input deck (i.e., 001 - 999[14]). The ***control_variable_number*** variable identifies the control component in the RELAP input deck associated with the channel. The card number is a three digit (or four digit if the expanded option is used) integer of the form ***nnn*** where ***nnn*** is equal to the control variable number. For example, the control***_variable_number*** for the control variable defined by card 205**535**00 is simply 535.

units: Format: String. Range: any. Spaces must not be used in the ***units*** variable (the underbar character "_" may be substituted for a space when needed). This variable specifies the units associated with the control variable quantity. The analyst should be aware that thermal hydraulic quantities derived from control variables will be in SI units unless the associated control variable card converts the quantity to another measurement system.

4. Sample Input

```
Control_volume         2
HV_204 204040000
HV_953 953010000

Control_junction       3
HJ_527 527000000
HJ_627 627000000
HJ_727 727000000

Variable_trips         1
VT_0465 465

Logical_trips          1
LT_1507 1507

Interactive_controls   4
IC_801 801
IC_802 802
IC_803 803
IC_804 804

Heat_structure    1
HS_57001 5701001  8
```

[14] An optional expanded format for control variables can also be enabled. This expands the allowable control variable range to 1 – 9999.

```
Control_variables       8
CV_002 002 Seconds
CV_100 100 %
CV_101 101 F
CV_477 477 Watts
CV_515 515 gpm
CV_519 519 gallons
CV_984 984 lbm/s
CV_864 864 rpm
```

This sample file provides examples of each type of RELAP Channels. The associated RELAP input deck for the problem must be consistent with these channel definitions or errors will be generated during input file processing.

1. Purpose

The SystemReliability.txt input file is used to identify conditional hardware failure events. A conditional hardware failure event is triggered when the associated component state is in the "ON" state (integer code 3022, "VON"). The failure state of the hardware control is specified by the controller ***failure_value*** or ***failure_state*** variable in the "ControlPanel.txt" input file. The analyst should specify failure events in the SystemReliability.txt input file to generate on-demand failure and success branches or to activate a failure event based on the status of a specified component state. Failures specified in the "SystemReliability.txt" file are not time dependent and are generated only when the associated component state is activated. The "Initiating_Event.txt" input file should be used to generate a single time-dependent failure branch (i.e., initiating events are actuated based on the elapsed simulation time and generate only an equipment failure branch).

When a conditional hardware failure event is activated, two event sequence branches are generated – a success branch and a failure branch. The operator may attempt to manual recover a failed component through the use of appropriate manual actions. If a recovery is attempted, two additional sequence branches are generated – a successful recovery branch where component function is fully restored and a permanent failure branch. Thus, each conditional hardware failure event can generate up to three branches: (1) no failure, (2) failure followed by successful recovery, and (3) permanent (unrecoverable) failure.

2. Input File Format

```
System_Reliability      number_of_reliability_events
component_state_name_1  control_name_1
alpha_fail          beta_fail          alpha_rec          beta_rec
.                   .                  .                  .
.                   .                  .                  .
component_state_name_n  control_name_n
alpha_fail          beta_fail          alpha_rec          beta_rec
```

3. Input Description

number_of_reliability_events: Format: Integer. Range: ≥ 0. Specifies the number of conditional hardware failure events provided in the input file. Failure events are read by the program in the order that they are listed in the input file. Therefore, if more than the specified ***number_of_reliability_events*** are described in the SystemReliablity.txt input file, only the first ***number_of_reliability_events*** are read.

Changing the ***number_of_reliability_events*** variable provides a convenient method for enabling and disabling failure events. The analyst can disable all reliability events by setting the ***number_of_reliability_events*** variable to 0 (it is not necessary to delete each failure event from the file). If only one failure event will be activated, the desired event is simply moved to the beginning of the failure event list and the ***number_of_reliability_events*** variable is set to 1 (all other reliability events will be ignored and need not be removed from the input file).

component_state_name: Format: String. Range: The ***component_state_name*** must match a control panel component state listed in the "ControlPanel.txt" input file. Specifies the component state that will be used to activate the conditional hardware failure event. Spaces must not be used in the ***component_state_name*** (the underbar character "_" may be substituted for a space when needed).

control_name: Format: String. Range: The ***control_name*** must match a control panel controller listed in the "ControlPanel.txt" input file. Specifies the controller associated with the failed hardware component. Spaces must not be used in the ***control_name*** (the underbar character "_" may be substituted for a space when needed). When the failure mode is activated, the control value for the controller will be set to the associated ***failure_value*** or ***failure_state***[15]**.**

alpha_fail: Format: Double. Range: > 0.0. Specifies the α parameter in the component failure distribution. To capture the uncertainty associated with failure estimates, the Beta Distribution is used to model the probability of hardware failure (Equation 19):

$$p(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{(\alpha-1)} (1-x)^{(\beta-1)}$$

$$\overline{p} = \frac{\alpha}{\alpha+\beta} \qquad (mean\ value)$$

(Equation 19)

Two parameters must be specified, the alpha parameter (α) and the beta parameter (β). During a simulation run, the failure probability is determined by a Monte Carlo sample of the failure distribution. To improve the reproducibility of simulation results, it is recommended that the α and β parameters be selected to minimize the variance of the failure distribution. For example, the failure distributions p(x, α = 1. β = 1.) and p(x, α = 100. β = 100.) both have the same mean value (0.5), but the variance for the latter case is more than an order of magnitude lower. A smaller variance will yield more reproducible simulation results.

beta_fail: Format: Double. Range: > 0.0. Specifies the β parameter in the component failure distribution.

---

15 The ***failure_value*** is used for controllers with fine adjustment capability. The ***failure_state*** applies to binary state controllers.

`alpha_rec:` Format: Double. Range: > 0.0. Specifies the α parameter in the component recovery distribution. After a conditional component failure is activated, the operator may attempt to return the equipment to a functional status by performing manual actions. An operator may attempt to recovery a failed component one time. If a recovery attempt is made, two sequence branches are generated: one branch for successful recovery and one branch for a permanent (unrecoverable) failure. To capture the uncertainty associated with recovery estimates, the Beta Distribution is used to model the recovery probability (Equation 20):

$$p(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{(\alpha-1)} (1-x)^{(\beta-1)}$$

$$\overline{p} = \frac{\alpha}{\alpha+\beta} \quad (\textit{mean value}) \qquad \text{(Equation 20)}$$

Two parameters must be specified, the alpha parameter (α) and the beta parameter (β). During a simulation run, the recovery probability is determined by a Monte Carlo sample of the failure distribution. To improve the reproducibility of simulation results, it is recommended that the α and β parameters be selected to minimize the variance of the failure distribution. For example, the failure distributions p(x, α = 1. β = 1.) and p(x, α = 100. β = 100.) both have the same mean value (0.5), but the variance for the latter case is more than an order of magnitude lower. A smaller variance will yield more reproducible simulation results.

In some cases, a recovery from a failure event is not physically possible or realistic. For example, the operators would not be able to recover from a steam generator tube rupture by simply "turning off" the rupture flow by resetting the control value for the appropriate control. Because recovery parameters must be specified for all failure events, it is recommended that the analyst use either easily recognizable dummy values for the alpha and beta parameters (e.g., 1.0) or set the recovery probability at a very low value (e.g., α = 1.0 and β = 10^6).

`beta_rec:` Format: Double. Range: > 0.0. Specifies the β parameter in the component recovery distribution.

4. <u>Sample Input</u>

```
System_Reliability      2
Main_Steam_Isolation    X_SGTR_SG_A
50000.          5.0          1.0        100000.0
Reactor_Trip            X_MD_AFW_Pump_A
100000.0        1.0          1.0        9.0
```

This sample file identifies two conditional hardware failure reliability events: (1) initiation of a steam generator tube rupture in the A SG (X_SGTR_SG_A), and (2) failure of the A train motor driven auxiliary feedwater pump (X_MD_AFW_Pump_A). Because the ***SystemReliablity.txt*** input file does not specify a failure control value, the component failure state must be specified in the ***ControlPanel.txt*** input file. For this example, the ***ControlPanel.txt*** file includes the following information:

Control Name	Interactive Control	Time	ON	OFF	Neutral	Failure Value
X_SGTR_SG_A	IC_836	1.0	**1.0**	-1.0	-1.0	**ON**
X_MD_AFW_Pump_A	IC_828	1.0	1.0	**0.0**	-1.0	**OFF**

Thus, the steam generator tube rupture event (X_SGTR_SG_A) failure state is specified as "ON" (control value of 1.0), and the motor driven auxiliary feedwater pump (X_MD_AFW_Pump_A) failure state is specified as "OFF" (control value of 0.0). The analyst must include appropriate logic in the RELAP input deck to support these events (see Section 2). When the "Main_Steam_Isolation" component state is first set to "ON" (integer code 3022, "VON"), two sequence branches will be generated. The first sequence branch will not include the failure (i.e., no steam generator tube rupture occurs). The second sequence branch will set variable IC_836 to 1.0 (which will initiate a steam generator tube rupture in the RELAP thermal hydraulic model) with an average branch probability of 0.9999[16]. Similarly, when the "Reactor_Trip" component state is set to "ON", one branch will be generated where no failure occurs and a second branch generated where interactive control variable IC_828 will be set to 0.0 (which will cause the A train motor driven auxiliary feedwater pump to fail). The operator may attempt to recovery the failed auxiliary feedwater pump with an average recovery probability of 0.1. System reliability conditional failure events are only actuated a single time during a simulation run.

[16] The average branch probability is calculated from the alpha and beta factors as follows: p = 50000 / (50000 + 5) ~ 0.9999.

General Description:

The ***ZCtrlPar.txt*** and ***ZiniADS.txt*** input files provide critical simulation control information for ADS-IDAC. The ***ZCtrlPar.txt*** file is used to specify the maximum time and probability cutoff truncation criteria. The ***ZiniADS.txt*** is used to identify the RELAP input deck that will be used for the simulation and the location of ADS-IDAC input files.

Input Files:

ZCtrlPar.txt
ZiniADS.txt

1. Purpose

The ***ZCtrlPar.txt*** input file is used to specify the sequence time and probability truncation criteria that will be used for the simulation. These truncation criteria can be set by the analyst to minimize the generation of low probability sequences and limit the simulation time.

2. Input File Format

```
CtrlPar
Truncation_probability        truncation_probability
Truncation_time               simulation_end_time
```

3. Input Description

truncation_probability: Format: double. Range: > 0.0. Specifies the probability threshold for sequence truncation. If the sequence probability decreases below the ***truncation_probability,*** the sequence is terminated.

simulation_end_time: Format: double. Range: > 0.0. The ***simulation_end_time*** is specified in seconds and must be less than or equal to the maximum RELAP end time entered in the input deck control cards (200 series cards). The analyst should be aware that this time is the maximum simulation time for a sequence – the sequence may be terminated prior to this time if certain conditions are encountered. Conditions that may terminate a sequence prior to the ***simulation_end_time*** include probability truncation, actuation of an "A_ENDSEQ_" alarm, or a "VSTOP" response not obtained action.

4. Sample Input

```
CtrlPar
Truncation_probability        1.0E-4
Truncation_time               6000.0
```

This sample input identifies a sequence probability truncation value of 10^{-4} and a sequence simulation time of 6000.0 seconds.

1. Purpose

The ***ZiniADS.txt*** is used to specify the path name location for the project input directory and the RELAP input deck thermal hydraulic model. This file must be located in the same directory as the executable version of ADS-IDAC (if the code will be run directly from the executable) or the same directory as the C++ workspace .dsw file (if ADS-IDAC will be run from the MS Visual C++ environment).

2. Input File Format

```
Project_Directory       project_directory_location
Input_Deck_File         RELAP_input_file
Simulation_Type         simulation_type
```

3. Input Description

project_directory_location: Format: string. Range: any. Specifies the path name location of the project input directory (i.e., the directory that contains the "input" and "output" subdirectories for the project. Quotation marks should not be used to delineate the string and the string must be provided on a single input line. The directory location must refer to an existing file directory and conform to standard Windows file directory naming conventions. It is recommended that the analyst use short directory names within the path name and the overall path name length be a concise as possible. The total string length must not exceed 80 characters or an error will occur.

RELAP_input_file: Format: string. Range: any. Specifies the path name and file name of the RELAP input deck (the ".i" file). Quotation marks should not be used to delineate the string and the string provided on a single line. The directory location must refer to an existing file directory and conform to standard Windows file directory naming conventions. It is recommended that the analyst use short directory names within the path name and the combined path and file name length be a concise as possible. The total string length must not exceed 80 characters or an error will occur.

simulation_type: Format: integer. Range: any. This variable is not currently used. A value of 0 is recommended.

4. Sample Input

```
Project_Directory C:\ADS_Client\TestCases\Generic_PWR
Input_Deck_File   C:\ADS_Client\ADSSource\Generic_PWR.i
Simulation_Type   0
```

This sample input identifies the “Generic_PWR” subdirectory as the location of the appropriate input files and specifies the “Generic_PWR.i” as the RELAP input deck. It is recommended that concise, but meaningful, names be used to identify project related directories and files.

Section 5: Term Conversions

```
// TermConversions.h: interface for the TermConversions class.
//
//////////////////////////////////////////////////////////////////////

#if
!defined(AFX_TERMCONVERSIONS_H__61704B1A_AF4F_44FA_8C5C_665C1B85916F__I
NCLUDED_)
#define
AFX_TERMCONVERSIONS_H__61704B1A_AF4F_44FA_8C5C_665C1B85916F__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
//
//consequences
#define CONSEQUENCE_BASE         -1
#define VNOKB                    -1
#define VCONS_SUCCEED             0
#define VTIME_TRUNCATED           1
#define VPROBABILITY_TRUNCATED    2
#define VTH_FAIL                  3 //RELAP5 crashed
#define VCREW_FAIL                4
#define VNON_RESPONSE_ACTION      5
#define VSG_OVERFED               6
#define VADS_SIM_ERR              7
#define VENDSEQ_ALARM             8
//Types of RELAP5 input deck component
#define VHYDRAULIC_VOLUME     1
#define VHYDRAULIC_JUNCTION   2
#define VVARIABLE_TRIP        3
#define VLOGICAL_TRIP         4
#define VINTERACTIVE_VARIABLE 5
#define VCONTROL_VARIABLE     6
#define VHEAT_STRUCTURE       7
//
//Types of Display for retrieving RELAP5 information to control panel
#define VDISPLAY_PARAMETER_VALUE          1
#define VDISPLAY_COMPONENT_STATE          2
#define VCONTROL_STATE                    3
#define VCONTROL_VALUE                    4
#define VALARM_COMPONENT                  5
#define VALARM_PARAMETER                  6
#define VALARM_TWO_PARAMETER_DIFFERENCE   7
#define VDISPLAY_HEAT_STRUCTURE           8
#define VDISPLAY_PARAMETER_TREND          9
//
//Types of communication
#define VALARM_STATE          1
#define VCOMPONENT_STATE      2
#define VPARAMETER_VALUE      3
#define VPARAMETER_TREND      4  //increasing, decreasing, stable
#define VPARAMETER_STATE      5  //too high, too low, high, low
#define VPARAMETER_DIFFERENCE 6
#define VCONTROL_COMMAND      7
#define VCONTROL_REPORT       8
//
//
```

```
#define VBASE            3000
#define VFALSE           3000
#define VTRUE            3001
#define VYES             3002
#define VNO              3003
#define VCHECKED         3004
#define VUN_CHECKED      3005
#define VGT              3006
#define VGE              3007
#define VEQ              3008
#define VLE              3009
#define VLT              3010
#define VDECREASING      3011
#define VINCREASING      3012
#define VSTABLE          3013
#define VBYPASS          3014
#define VAG_OPERATOR     3015
#define VPR_PANEL        3016
#define VPR_OPERATOR     3017
#define VPR_PLANT        3018
#define VNORMAL          3019
#define VFAILED          3020
#define VSUCCEED         3021
#define VON              3022
#define VOFF             3023
#define VRELIABILITY     3024
#define VPANEL           3025
#define VPLANT           3026
#define VACTIVATED       3027
#define VSKIP            3028
#define VSTANDBY         3029
#define VOPERATE         3030
#define VCONTROL         3031
#define VBETWEEN         3032
#define VAUTO            3033
#define VRESET           3034
#define VNEW             3035
#define VFAILED_P        3036
#define VODM_ASK         3037
#define VODM_COM         3038
#define VODM_ADV         3039
#define VOAT_SEND        3040      //action taker sends information
#define VOCT_ADV         3041      //consultant's advise
#define VOAT             3042      //operator action taker
#define VODM             3043      //operator decision maker
#define VOCT             3044      //operator consultant
#define VNONE            3045
#define VGNOP            3046      //goal of normal operation
#define VGTS             3047      //goal of trouble shooting
#define VGMGSM           3048      //goal of maintain global safety
#define VGMON            3049      //goal of monitoring abnormal cond
#define VGMESM           3050      //goal of maintain equipment safety
#define VSGFRC           3051      //subgoal of find root cause
#define VSGDAP           3052      //subgoal of decide a action
#define VSGPA            3053      //subgoal of performing a action
#define VSWM             3054      //strategy of wait and monitor
#define VSDM             3055      //strategy of direct matching
```

```
#define VSLR               3056     //strategy of limited reasoning
#define VSIR               3057     //strategy of instinctive response
#define VSIDR              3058     //strategy of inductive and
                                    //deductive reasoning
#define VSFP               3059     //strategy of follow procedure
#define VSTE               3060     //strategy of trial and error
#define VSAA               3061     //strategy of ask advise
#define VSFI               3062     //strategy of following instruction
#define VSRTA              3063     //report taken action
#define VSRPI              3064     //report perceived information
#define VSGA               3065     //giving advise
#define VSGC               3066     //strategy of give command (ODM)
#define VSAI               3067     //ask information (ODM)
#define VGOAL              3068     //goal
#define VSUBGOAL           3069     //subgoal
#define VSTRATEGY          3070     //strategy
#define VDIAGNOSIS         3071     //diagnosis
#define VACTIONOBJECTIVE   3072     //action task
#define VACTIONPACK        3073     //action package
#define VACTION            3074     //action
#define VACTIONC           3075     //check command
#define VACTIONAC          3076     //action command
#define VACTIONA           3077     //direct action
#define VCONFIRMC          3078     //the expectation check when
                                    //subgoal is decide action package
#define VACTIONI           3079     //the OAT send info to ODM
#define VACTIONIA          3080     //the OCT gives advise to ODM
#define VPROCESSING        3081     //processing for index
#define VPROCESSING_A      3082     //processing and data downloaded
#define VLOCALE            3083     //local equipment problem
#define VLOCALP            3084     //local parameter problem
#define VGLOBAL            3085     //global problem
#define VREPLACE           3086     //information merge type : replace
#define VCOMBINE           3087     //information merge type : combine
#define VNOCHANGE          3088     //information merge type : no
                                    //change
#define VINTERRUPTED       3089
#define VWAITING           3090
#define VOLDINFO           3091
#define VPROCEDURE_XFR     3092     //flag for transfers between mental
                                    //& paper procedures
#define VSELF              3093
#define VTRANSIENT         3094
#define VBRANCH            3095
#define VES                3096
#define VDONE              3097
#define VOPEN              3098
#define VCLOSE             3099
#define VEXPECT_CONTROL    3100
#define VDIAALARM          3101
#define VDIACOMP           3102
#define VDIAPAR            3103
#define VDIACOMBINE        3104
#define VFAIL_TO_START     3105
#define VFAIL_TO_RUN       3106
#define VFAIL_TO_OPEN      3107
#define VFAIL_TO_CLOSE     3108
```

```
#define VVERIFY            3109
#define VPROCEDURE         3110
#define VPROPORTIONAL      3111
#define VADDITIVE          3112
#define VSLEADV            3113     //advise
#define VSPECIAL           3114
#define VTRIP              3115
#define VSTOP              3116
#define VTRPVLV            3117
#define VSRVVLV            3118
#define VTERMINATE_IMA     3119
#define VGREATER_THEN_ON   3120
#define VSMALLER_THEN_ON   3121
#define VGREATER_THEN_OPEN 3122
#define VSMALLER_THEN_OPEN 3123
#define VUNKNOWN           3124
#define VMENTAL            3125
#define VACTIVE            3126
#define VPAUSE             3127
#define VPROCEDURE_TRANSFER  3128
#define VINFO_GATHER_MODE    3129
#define VMENTAL_BELIEF       3130
#define VRECOVERY            3131
#define VGREATER_THEN_OFF    3132
#define VSMALLER_THEN_OFF    3133
#define VGREATER_THEN_CLOSE  3134
#define VSMALLER_THEN_CLOSE  3135
#define VOAT_SCAN            3136   //operator action taker scan
                                    //parameters
#define VODM_SCAN            3137   //operator decision maker scan
                                    //parameters
#define VOCT_SCAN            3138   //operator consultant scan
                                    //parameters
#define VPARAMETER_CONTROL   3139   //set action control value to
                                    //parameter value
#define VPARAMETER_SCAN      3140   //Parameter scan
#define VNOP                 3151   //normal operating procedure
#define VAOP                 3152   //abnormal operating procedure
#define VEOP                 3153   //emergency operating procedure
#define VFRG                 3154   //functional recovery guideline
                                    //procedure
#define VECA                 3155   //emergency contingency action
#define VMENTAL_PROCEDURE    3156   //mental procedure step
#define VOBJECTIVE_RELATED   3161   //objective related action
                                    //procedure step
#define VPREREQUISITE_ACTION 3162   //prerequisite related procedure
                                    //action step
#define VMONITORING_STEP     3163   //monitoring procedure step
#define VVERIFICATION_STEP   3164   //verification procedure step
#define VDIAGNOSIS_STEP      3165   //diagnosis/decision related
                                    //procedure step
#endif //
!defined(AFX_TERMCONVERSIONS_H__61704B1A_AF4F_44FA_8C5C_665C1B85916F__I
NCLUDED_)
```

(this page intentionally blank)

Section 6: References

Chang, Y. H., Coyne, K., & Mosleh, A. (2006). A Nuclear Plant Accident Diagnosis Method to Support Prediction of Errors of Commission. Paper presented at the American Nuclear Society Nuclear Plant Instrumentation, Control and Human-Machine Interface Technology Conference (NPIC & HMIT 2006), Albuquerque, New Mexico.

Chang, Y. H., & Mosleh, A. (2007a). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents - Part 1: Overview of the IDAC Model. Reliability Engineering & System Safety, 92, 997-1013.

Chang, Y. H., & Mosleh, A. (2007b). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents - Part 2: IDAC performance influencing factors model. Reliability Engineering & System Safety, 92, 1014–1040.

Chang, Y. H., & Mosleh, A. (2007a). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents Part 3: IDAC operator response model. Reliability Engineering and System Safety, 92, 1041-1060.

Chang, Y. H., & Mosleh, A. (2007b). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents Part 5: Dynamic probabilistic simulation of the IDAC model. Reliability Engineering and System Safety, 92, 1076-1101.

Chang, Y. H., & Mosleh, A. (2007c). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents. Part 4: IDAC causal model of operator problem-solving response. Reliability Engineering and System Safety, 92, 1061-1075.

Coyne, K., & Mosleh, A. (2008a, May 18-25, 2008). Implementation of a Dynamic PRA Approach for the Prediction of Operator Errors During Abnormal Nuclear Power Plant Events. Paper presented at the Ninth International Probabilistic Safety Assessment and Management Conference (PSAM-9), Hong Kong.

Coyne, K., & Mosleh, A. (2009, September 7-10, 2009). Modeling nuclear power plant operator knowledge and procedural adherence in the ADS-IDAC simulation model. Paper presented at the European Safety and Reliability Association (ESRA) and Society for Risk Analysis Europe (SRA-E) Annual Conference (ESREL 2009), Prague, Czech Republic.

Hsueh, K. S., & Mosleh, A. (1993, March 29 - April 1 , 1993,). ADS; accident dynamic simulator for nuclear power plant accident analysis. Paper presented at the Simulators X: 1993 Simulation Multiconference on the International Simulators Conference, Arlington, VA.

Hsueh, K. S., & Mosleh, A. (1996). The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants. Reliability Engineering & System Safety, 52(3), 297-314.

Hsueh, K. S., Mosleh, A., Sharon, A., & Sharon, A. (1993). ADS: accident dynamic simulator for nuclear power plant accident analysis. In Simulators X. Proceedings of the 1993 Simulation Multiconference on the International Simulators Conference (pp. 103-108). San Diego, CA: SCS.

Hsueh, K. S., Soth, L., Mosleh, A., Apostolakis, G. E., Wu, J. S., et al. (1994). A simulation study of errors of commission in nuclear accidents. In Proceedings of PSAM - II. An International Conference Devoted to the Advancement of System-based Methods for the Design and Operation of Technological Systems and Processes (Vol. 2, pp. 066/061-066). Los Angeles, CA.

Mosleh, A., & Chang, Y. H. (2004). Model-based human reliability analysis: prospects and requirements. Reliability Engineering and System Safety, 83, 241-253.

Mosleh, A., Smidts, C., & Shen, S. H. (1996). Modeling cognition dynamics and its application to human reliability analysis. Transactions of the American Nuclear Society, 75, 84-85.

Siu, N. (1990). Technical Note - Dynamic Accident Sequence Analysis in PRA: A Comment on 'Human Reliability Analysis-Where Shouldst Thou Turn?' Reliability Engineerirlg and System Safety, 29, 359-364.

Siu, N. (1994). Risk assessment for dynamic systems: An overview. Reliabilify Engineering and System Safety, 43, 43-73.

Smidts, C., Shen, S. H., & Mosleh, A. (1997). The IDA cognitive model for the analysis of nuclear power plant operator response under accident conditions. Part I: Problem solving and decision making model. Reliability Engineering & System Safety, 55(1), 51-71.

(this page intentionally blank)

Bibliography

[1] R. Boring, "Third Generation Human Reliability Analysis: Benefits and Challenges of Simulating Human Performance," presented at Enlarged Halden Programme Group Meeting, Storefjell, Norway, 2007.

[2] D. I. Gertman, B. P. Hallbert, M. W. Parrish, M. B. Sattision, D. Brownson, and J. P. Tortorelli, "Review of Findings for Human Error Contribution to Risk in Operating Events," U.S. Nuclear Regulatory Commission NUREG/CR-6753, 2001.

[3] NUREG, "Three Mile Island - A Report to the Commissioners and to the Public," US Nuclear Regulatory Commission NUREG/CR-1250, 1980.

[4] J. Reason, "Errors and Violations: the Lessons of Chernobyl," presented at IEEE Fourth Conference on Human Factors and Power Plants, Monterey, CA, 1988.

[5] J. V. Kauffman, "Engineering Evaluation - Operating Events With Inappropriate Bypass or Defeat of Engineered Safety Features," US Nuclear Regulatory Commission, Office of Analysis and Evaluation of Operational Data AEOD/E95-01, 1995.

[6] NUREG, "Individual Plant Examination Program: Perspectives on Reactors Safety and Plant Performance," US Nuclear Regulatory Commission NUREG-1560, 1997.

[7] NUREG, "PRA Procedures Guide," U.S. Nuclear Regulatory Commission NUREG/CR-2300, January 1983.

[8] E. Hollnagel, "Reliability analysis and operator modeling," *Reliability Engineering and System Safety*, vol. 52, pp. 327-337, 1996.

[9] A. D. Swain and H. E. Guttmann, "Handbook of Human Reliability Analysis With Emphasis on Nuclear Power Plant Applications," US Nuclear Regulatory Commission NUREG/CR-1278, 1983.

[10] M. Cunningham, M. Barriere, D. Bley, S. Cooper, J. Forester, A. Kolaczkowski, W. Luckas, G. Parry, A. Ramey-Smith, C. Thompson, D. Whitehead, and J. Wreathall, *A Technical Basis and Implementation Guidelines for a Technique for Human Event Analysis (NUREG-1624)*. Washington, DC: U.S. Nuclear Regulatory Commission, 2000.

[11] N. Siu, "Technical Note - Dynamic Accident Sequence Analysis in PRA: A Comment on 'Human Reliability Analysis-Where Shouldst Thou Turn?'" *Reliability Engineering and System Safety*, vol. 29, pp. 359-364, 1990.

[12] S. Grant, "Cognitive architecture for modelling human error in complex dynamic tasks," *Travail Humain*, vol. 60, pp. 363-385, 1997.

[13] G. Klein, "The Recognition-Primed Decision (RPD) Model: Looking Back, Looking Forward," in *Naturalistic Decision Making*, C. E. Zsambok and G. Klein, Eds. Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc., 1997, pp. 285-292.

[14] G. Klein and D. Klinger, "Naturalistic Decision Making," *Human Systems IAC GATEWAY*, vol. XI, pp. 16-19, 1991.

[15] J. Rasmussen, "Models of Mental Strategies in Process Plant Diagnosis," presented at NATO Symposium on Human Detection and Diagnosis of System Failures, Roskilde, Denmark 1980.
[16] Y. H. Chang, K. Coyne, and A. Mosleh, "A Nuclear Plant Accident Diagnosis Method to Support Prediction of Errors of Commission," presented at American Nuclear Society Nuclear Plant Instrumentation, Control and Human-Machine Interface Technology Conference (NPIC & HMIT 2006), Albuquerque, New Mexico, 2006.
[17] M. Lind, "The Use of Flow Models for Automated Plant Diagnosis," presented at NATO Symposium on Human Detection and Diagnosis of System Failures, Roskilde, Denmark 1980.
[18] M. Lind, "Plant Modeling for Human Supervisory Control," *Transactions of the Institute of Measurement and Control*, vol. 21, pp. 171-180, 1999.
[19] R. E. Mayer, *Thinking, Problem Solving, Cognition*, Second ed. New York, New York: W. H. Freeman and Company, 1992.
[20] Y. H. Chang and A. Mosleh, "Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents - Part 2: IDAC performance influencing factors model," *Reliability Engineering & System Safety*, vol. 92, pp. 1014–1040, 2007.
[21] G. Sax, *Principles of Educational and Psychological Measurement and Evaluation*, Second ed. Belmont, California: Wadsworth Publishing Company, 1980.
[22] Y. H. Chang and A. Mosleh, "Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents - Part 1: Overview of the IDAC Model," *Reliability Engineering & System Safety*, vol. 92, pp. 997-1013, 2007.
[23] C. Smidts, S. H. Shen, and A. Mosleh, "The IDA cognitive model for the analysis of nuclear power plant operator response under accident conditions. Part I: Problem solving and decision making model," *Reliability Engineering & System Safety*, vol. 55, pp. 51-71, 1997.
[24] A. Mosleh and Y. H. Chang, "Model-based human reliability analysis: prospects and requirements," *Reliability Engineering and System Safety*, vol. 83, pp. 241-253, 2004.
[25] E. M. Roth, R. J. Mumaw, and P. M. Lewis, "An Empirical Investigation of Operator Performance in Cognitively Demanding Simulated Emergencies," U.S. Nuclear Regulatory Commission NUREG-CR-6208, 1994.
[26] D. O'Hare, "Cognitive Functions and Performance Shaping Factors in Aviation Accidents and Incidents," *The International Journal of Aviation Psychology*, vol. 16, pp. 145-156, 2006.
[27] D. O'Hare, M. Wiggins, and R. Batt, "Cognitive failure analysis for aircraft accident investigation," *ERGONOMICS*, vol. 37, pp. 1855-1869, 1994.
[28] R. T. Clemen and T. Reilly, *Making Hard Decisions*, Second ed. Pacific Grove, CA: Duxbury Thomson Learning, 2001.
[29] H. Mintzberg, D. Raisinghani, and A. Théoret, "The Structure of 'Unstructured' Decision Processes," *Administrative Science Quarterly*, vol. 21, pp. 246, 1976.

[30] J. Reason, *Human Error*. New York, New York: Cambridge University Press, 1990.
[31] D. I. Gertman and H. S. Blackman, *Human Reliability & Safety Analysis Data Handbook*. New York, New York: John Wiley & Sons, Inc., 1994.
[32] J. Rasmussen, "Human Errors: A Taxonomy for Describing Human Malfunction in Industrial Installations," *Journal of Occupational Accidents*, vol. 4, pp. 311-333, 1982.
[33] J. Rasmussen, "Human Factors in High Risk Systems," presented at IEEE Fourth Conference on Human Factors and Power Plants, Monterey, CA, 1988.
[34] E. Hollnagel, "Looking for errors of omission and commission or The Hunting of the Snark revisited," *Reliability Engineering and System Safety*, vol. 68, pp. 135-145, 2000.
[35] A. Kolaczkowski, J. Forester, E. Lois, and S. Cooper, "Good Practices for Implementing Human Reliability Analysis," U.S. Nuclear Regulatory Commission NUREG-1792, 2005.
[36] D. A. Norman, *The Design of Everyday Things*, Reprint ed. New York, New York: Basic Books (Doubleday), 2002.
[37] C. Smidts, S.-H. Shen, and A. Mosleh, "A Taxonomy And Root-Cause Analysis Of Human Cognitive Behavior Based On A Cognitive Model," presented at Annual Reliability and Maintainability Symposium, 1995.
[38] A. Singh and A. J. Spurgin, "Plant Simulators Used to Measure Operator Crew Reliability," presented at International Symposium on Balancing Automation and Human Action in Nuclear Power Plants, Munich, Germany, 1990.
[39] E. M. Roth, D. D. Woods, and H. E. Pople, Jr., "Cognitive simulation as a tool for cognitive task analysis," *ERGONOMICS*, vol. 35, pp. 1163-1198, 1992.
[40] P. C. Cacciabue, "Cognitive modeling: a fundamental issue for human reliability assessment methodology?," *Reliability Engineering & System Safety*, vol. 38, pp. 91-97, 1992.
[41] F. Decortis, "Operator strategies in a dynamic environment in relation to an operator model," *ERGONOMICS*, vol. 36, pp. 1291-1304, 1993.
[42] M. D. Byrne and A. Kirlik, "Using Computational Cognitive Modeling to Diagnose Possible Sources of Aviation Error," *The International Journal of Aviation Psychology*, vol. 15, pp. 135-155, 2005.
[43] N. A. Taatgen, C. Lebiere, and J. R. Anderson, " Modeling Paradigms in ACT-R," in *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*, R. Sun, Ed.: Cambridge University Press, 2006, pp. 29-52.
[44] E. M. Altmann and J. G. Trafton, "Memory for goals: An activation-based model," *Cognitive Science*, vol. 26, pp. 39-83, 2002.
[45] A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, and U. Catalyurek, "Dynamic generation of accident progression event trees," *Nuclear Engineering and Design (2008)* vol. 238, pp. 3457-3467, 2008.
[46] E. Hofer, M. Kloos, B. Krzykacz-Hausmann, J. Peschke, and M. Sonnenkalb, "Dynamic Event Trees for Probabilistic Safety Analysis," in *Eurosafe - 2002*.

Berlin, Germany: Gesellschaft für Anlagen- und Reaktorsicherheit (GRS), 2002.
[47] M. Kloos and J. Peschke, "Consideration of human actions in combination with the probabilistic dynamics method MCDET " presented at ESREL 2007 Safety and Reliability Conference, Stavenger, Norway, 2007.
[48] A. Mosleh, C. Smidts, and S. H. Shen, "Modeling cognition dynamics and its application to human reliability analysis," *Transactions of the American Nuclear Society*, vol. 75, pp. 84-85, 1996.
[49] S. H. Shen, C. Smidts, and A. Mosleh, "Application of a Model Based Human Error Taxonomy to the Analysis of Reactor Operating Events," presented at Topical Meeting on Computer-Based Human Support Systems: Technology, Methods, and Future, Philadelphia, Pennsylvania, 1995.
[50] S. H. Shen, C. Smidts, and A. Mosleh, "A methodology for collection and analysis of human error data based on a cognitive model: IDA," *Nuclear Engineering and Design*, vol. 172, pp. 157-186, 1997.
[51] K. S. Hsueh and A. Mosleh, "The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants," *Reliability Engineering & System Safety*, vol. 52, pp. 297-314, 1996.
[52] D. Zhu, Y. H. Chang, and A. Mosleh, "The Use of Distributed Computing for Dynamic PRA: The ADS Approach," presented at Ninth International Probabilistic Safety Assessment and Management Conference (PSAM-9), Hong Kong, 2008.
[53] J. A. Swets, W. P. Tanner, and T. G. Birdsall, "Decision Processes in Perception," *Psychological Review*, vol. 68, pp. 301-340, 1961.
[54] I. Erev, "Signal Detection by Human Observers: A Cutoff Reinforcement Learning Model of Categorization Decisions Under Uncertainty," *Psychological Review*, vol. 105, pp. 280-298, 1998.
[55] R. Barkan, D. Zohar, and I. Erev, "Accidents and Decision Making under Uncertainty: A Comparison of Four Models," *ORGANIZATIONAL BEHAVIOR AND HUMAN DECISION PROCESSES*, vol. 74, pp. 118-144, 1998.
[56] K. J. Vicente, R. J. Mumaw, and E. M. Roth, "Operator monitoring in a complex dynamic work environment: a qualitative cognitive model based on field observations," *Theoretical Issues in Ergonomics Science*, vol. 5, pp. 359-384, 2004.
[57] R. Hastie and R. Dawes, *Rational Choice in an Uncertain World - The Psychology of Judgment and Decision Making*. Thousand Oaks, California: Sage Publications, Inc., 2001.
[58] G. Gigerenzer, U. Hoffrage, and H. Kleinbölting, "Probabilistic Mental Models: A Brunswikian Theory of Confidence," *Psychological Review*, vol. 98, pp. 506-528, 1991.
[59] P. Juslin, H. Olsson, and M. Bjorkman, "Brunswikian and Thurstonian Origins of Bias in Probability Assessment: On the Interpretation of Stochastic Components of Judgment," *Journal of Behavioral Decision Making*, vol. 10, pp. 189-209, 1997.

[60] A. Tversky and D. Kahneman, "Judgment Under Uncertainty: Heuristics and Biases," *Science*, vol. 185, pp. 1124-1131, 1974.
[61] G. Gigerenzer, "I Think, Therefore I Err," *Social Research*, vol. 72, pp. 195-218, 2005.
[62] G. Gigerenzer and P. Todd, *Simple Heuristics That Make Us Smart*. New York, New York: Oxford University Press, Inc., 1999.
[63] D. Kahneman and A. Tversky, "The Simulation Heuristic," in *Judgment under uncertainty: Heuristics and Biases*, D. Kahneman, P. Slovic, and A. Tversky, Eds. New York, New York: Cambridge University Press, 1982.
[64] G. Gigerenzer, "On Narrow Norms and Vague Heuristics: A Reply to Kahneman and Tversky (1996)," *Psychological Review*, vol. 103, pp. 592-596, 1996.
[65] A. Tversky and D. Kahneman, "Extensional Versus Intuitive Reasoning: The Conjunction Fallacy in Probability Judgment," *Psychological Review*, vol. 90, pp. 293-315, 1983.
[66] T. Wallsten, "The Theoretical Status of Judgment Heuristics," in *Decision Making Under Uncertainty*, R. W. Scholz, Ed. Amsterdam: North Holland Publishing Company, 1983, pp. 21-37.
[67] R. Lipshitz, G. Klein, J. Oransanu, and E. Salas, "Focus Article: Taking Stock of Naturalistic Decision Making," *Journal of Behavioral Decision Making*, vol. 14, pp. 331-352, 2001.
[68] C. E. Zsambok, "Naturalistic Decision Making: Where Are We Now?," in *Naturalistic Decision Making*, C. E. Zsambok and G. Klein, Eds. Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc., 1997, pp. 3-16.
[69] R. Lipshitz, "Naturalistic Decision Making Perspectives on Decision Errors," in *Naturalistic Decision Making*, C. E. Zsambok and G. Klein, Eds. Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc, 1997, pp. 151-160.
[70] B. M. Perrin, B. J. Barnett, L. Walrath, and J. D. Grossman, "Information Order and Outcome Framing: An Assessment of Judgment Bias in a Naturalistic Decision Making Context," *Human Factors*, vol. 43, pp. 227-238, 2001.
[71] D. A. Wiegmann, J. Goh, and D. O'Hare, "The Role of Situational Assessment and Flight Experience in Pilots' Decisions to Continue Visual Flight Rules into Adverse Weather," *Human Factors*, vol. 44, pp. 189-197, 2002.
[72] "RELAP5/MOD3 CODE MANUAL," U.S. Nuclear Regulatory Commission NUREG/CR-5535 (INEL-95/0174), 1995.
[73] N. Siu, "Risk assessment for dynamic systems: An overview," *Reliability Engineering and System Safety*, vol. 43, pp. 43-73, 1994.
[74] R. J. Mumaw, I. E. Schoenfeld, and J. J. Persensky, "Training Cognitive Skills for Severe Accident Management," pp. 374-380, 1993.
[75] R. J. Mumaw, E. M. Roth, K. J. Vicente, and C. M. Burns, "There Is More to Monitoring a Nuclear Power Plant than Meets the Eye," *Human Factors*, vol. 42, pp. 36-55, 2000.
[76] W. Frid, M. Knochenhauer, and M. Bednarski, "Development of a Bayesian belief network for a boiling water reactor during fault conditions," *Computer Assisted Mechanics and Engineering Sciences*, vol. 12, pp. 133-145, 2005.

[77] C. Kang and M. Golay, "A Bayesian belief network-based advisory system for operational availability focused diagnosis of complex nuclear power systems," *Expert Systems with Applications*, vol. 17, pp. 21-32, 1999.
[78] M. Zavisca, H. Kahlert, and M. Khatib-Rahbar, "A Bayesian Network Approach to Accident Management and Estimation of Source Terms for Emergency Planning," presented at PSAM7/ESREL'04 Conference, Berlin, Germany, 2004.
[79] M. R. Lee, "Expert System for Nuclear Plant Accident Diagnosis Using a Fuzzy Inference Method," *Expert Systems*, vol. 19, pp. 201-207, 2002.
[80] Y. Maruyama and R. Takahashi, "Practical expression for exception and its utilization to leakage identification of power plant," *Fuzzy Sets and Systems*, vol. 34, pp. 1-13, 1990.
[81] M. Shahinpoor and D. J. Wells, "Applications and possibilities for fuzzy failure analysis and diagnosis of reactor plant components and areas," *Nuclear Engineering and Design*, vol. 61, pp. 93-100, 1980.
[82] "Intentional Bypassing of Automatic Actuation of Plant Protective Features," U.S. Nuclear Regulatory Commission, Information Notice 92-47, 1992.
[83] "Reduction in Reactor Coolant system Pressure Due to a Failure of a Pressurizer Spray Valve and Associated Position Indication Results in Actuation of Reactor Protection System and Engineered Safeguards," U.S. Nuclear Regulatory Commission Docket #50-302, Licensee Event Report 91-018-00 (ADAMS Accession No. ML9201130206), 1992.
[84] "Trip Report: Onsite Analysis of the Human Factors of an Event at Crystal River Unit 3 December 8, 1991," U.S. Nuclear Regulatory Commission, EGG-HFRU-10085 (ADAMS Accession No. ML9202060214), 1992.
[85] J. Forester, A. Kolaczkowski, S. Cooper, D. Bley, and E. Lois, "ATHEANA User's Guide," U.S. Nuclear Regulatory Commission NUREG-1880, 2007.
[86] "ANS 3.2/ANSI N18.7," Administrative Controls and Quality Assurance for the Operational Phase of Nuclear Power Plants, 1976.
[87] J. Park and W. Jung, "The operators' non-compliance behavior to conduct emergency operating procedures—comparing with the work experience and the complexity of procedural steps," *Reliability Engineering and System Safety*, vol. 82, pp. 115-131, 2003.
[88] D. Gertman, H. Blackman, J. Marble, J. Byers, and C. Smith, "The SPAR-H Human Reliability Analysis Method," US Nuclear Regulatory Commission, Washington, DC NUREG/CR-6883, 2005.
[89] "Task Action Plan item I.D.2 - Safety Parameter Display System," US Nuclear Regulatory Commission Generic Letter 89-06, 1989.
[90] H. Pesme, P. Le Bot, and P. Meyer, "Little Stories to Explain Human Reliability Assessment: a practical approach of the MERMOS method," presented at Joint 8th Human Factors and Power Plants (HFPP) and Human Performance/Root Cause/Trending (HPRCT) 13th Annual Meeting Monterey, California, 2007.
[91] E. Lois, V. N. Dang, J. Forester, H. Broberg, S. Massaiu, M. Hildebrandt, P. Ø. Braarud, G. Parry, J. Julius, R. Boring, I. Männistö, and A. Bye, "Halden Work Report: International HRA Empirical Study - Pilot Phase Report

Description of Overall Approach and First Pilot Results from Comparing HRA Methods to Simulator Data," OECD Halden Reactor Project HWR-844, 2008.
[92] E. Lois, V. N. Dang, G. Parry, J. Forester, S. Massaiu, R. Boring, P. Ø. Braarud, H. Broberg, J. Julius, I. Männistö, and A. Bye, "THE INTERNATIONAL HRA EMPIRICAL STUDY - RESULTS FROM COMPARING HRA METHODS PREDICTIONS TO HAMMLAB SIMULATOR DATA ON SGTR SCENARIOS," OECD Halden Reactor Project HWR-915 (DRAFT Report), 2009.
[93] D. R. Krathwhol, *Methods of Educational and Social Science Research*. White Plains, New York: Longman, 1993.
[94] B. Hallbert and A. Kolaczkowski, "The Employment of Empirical Data and Bayesian Methods in Human Reliability Analysis: A Feasibility Study," U.S. Nuclear Regulatory Commission NUREG/CR-6949, 2007.
[95] B. Hallbert, R. Boring, D. Gertman, D. Dudenhoeffer, A. Whaley, J. Marble, J. Joe, and E. Lois, "Human Event Repository and Analysis (HERA) System, Overview," U.S. Nuclear Regulatory Commission NUREG/CR-6903, 2006.
[96] Y. H. Chang and A. Mosleh, "Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents Part 3: IDAC operator response model," *Reliability Engineering and System Safety*, vol. 92, pp. 1041-1060, 2007.
[97] K. Furuta, Y. Soraji, T. Kanno, H. Aoyama, S. Inoue, D. Karikawa, and M. Takahashi, "Cognitive model of team cooperation in en-route air traffic control," presented at European Safety and Reliability Association Annual Conference - ESREL-2009, Prague, Czech Republic, 2009.
[98] S. Burton, C. Phillips, P. Snyder, and R. Ng, "Indiana Michigan Power Company; Inspection Procedure 95002, "Inspection For One Degraded Cornerstone Or Any Three White Inputs In A Strategic Performance Area". ," U.S. Nuclear Regulatory Commission Inspection Report No. 05000316-04-004, 04/19/2004 - 05/12/2004 2004.
[99] L. Jaffe, W. M. J. Bland, R. English, R. M. Eytchison, D. Reilly, W. Stratton, and J. L. Tew, "THE PRESIDENT'S COMMISSION ON THE ACCIDENT AT THREE MILE ISLAND: REPORTS OF THE TECHNICAL ASSESSMENT TASK FORCE ON TECHNICAL STAFF ANALYSIS REPORTS SUMMARY - SUMMARY SEQUENCE OF EVENTS," Presidents Commission on the Accident at Three Miles Island, Washington, D.C. October 1979.
[100] V. N. Dang and A. Bye, "Evaluating HRA Methods in Light of Simulator Findings: Study Overview and Issues for an Empirical Test," presented at Enlarged Halden Programme Group Meeting, Storefjell, Norway, 2007.
[101] K. Coyne and A. Mosleh, "ADS-IDAC Analysis of Loss of Feedwater Accident Scenarios - OECD Halden Reactor Project International HRA Empirical Study," University of Maryland, Center for Risk and Reliability August 30, 2008.