AN INTEGRATED SYSTEM FOR COMPUTER BASED TRAINING OF PROCESS OPERATORS

by

Symeon Christofer KASSIANIDES
B.Sc.Ch.Eng. (M.I.T.)

June 1991

A thesis submitted for the degree of Doctor of Philosophy of the University of London
and for the Diploma of Membership of the Imperial College

Department of Chemical Engineering and Chemical Technology
Imperial College of Science, Technology and Medicine
London SW7

ABSTRACT

To perform their tasks, operators refer to different types of knowledge, employ a variety of skills and perform a range of cognitive activities. Furthermore, as their role evolves training requirements become increasingly complex and sophisticated. These in turn demand advanced and adaptable training methods and training delivery platforms. Computer based training (CBT) offers these possibilities.

Although a great range of CBT systems have been developed, their routine use by the process industries, especially of training simulators, is presently limited due mainly to the effort and the costs involved in the initial development of CBT systems and courseware. As a result these are generally tailored to specific problems and are often deeply embedded in bespoke hardware and software implementations. Subsequent maintenance and changes are therefore difficult, leading to entire systems becoming rapidly obsolete. This is especially true in the case of training simulators where major cost factors are identified in the initial development of models for plant and control systems, robust and fast numerical solution techniques for simulation in real-time, operational and training sequences, and sophisticated user interfaces.

A need therefore exists for tools that will allow the easy development of flexible and maintainable CBT systems and courseware. While a range of CBT authoring systems abound, none specifically meets the process industries general training requirements and its particular need for advanced simulation capabilities.

A second area of interest is the need for stand-alone CBT systems, especially ones that integrate theory and practice through the integration of qualitative and quantitative training methods. It is currently difficult to achieve this and in particular to merge dynamic simulation with tutorial methods. Very little is currently available in this respect.

A third area involves the formal definition of training programmes for CBT implementations. While a range of training programmes have been formally described, their implementation in CBT systems is generally not addressed. A special case is training for fault diagnosis. While an active research area, little has been done in formulating training programmes that may be implemented stand-alone in a CBT system.

A novel integrated architecture for CBT of process operators is proposed. Based on this an integrated system design was developed and implemented which achieves advanced operator training functionalities and is applicable to a wide range of training purposes in the process industries. The system attempts to minimize the user

development effort through the integration of high level, general purpose modules: a commercial dynamic simulator; a plant control system; a supervisory system for planning and management of discrete operations; and an expert system module for the definition of training programmes and their coordination with the other modules.

Within the system, a variety of plant models, operating procedures and training scenarios are easily defined. A range of training methods may be incorporated, including structured tutorials, guided steady-state and dynamic demonstrations of typical operations, fully interactive plant operation from a variety of control viewpoints, and training for fault diagnosis with provision of advice. Thus, the proposed system provides training support tools for the easy development, update, modification and maintenance of training programmes with the possibility to focus on either the qualitative or the quantitative approaches to training or to combine the two.

Process Trainer, a prototype 'expert system shell' for process training applications, illustrates the above features and functionalities. Three training programmes have been implemented within *Process Trainer*, each addressing a different training area. They employ a variety of training methodologies, are stand-alone while also permitting on-line trainer intervention, and deliver instruction through a combination of theory and practice. Furthermore, the formal methodological definition of *Process Trainer* permits the easy definition and integration of additional training programmes.

Two of the training programmes are a novel implementation of formalized training programmes implementing the structured approach suggested by Bainbridge (1990). The *Tutorial* aims to prepare an operator for plant operation while the *Plant Operation Training Session* places emphasis on actual real-time plant operation. A novel strategy for fault diagnosis is proposed which forms the basis for the third *Training for Fault Identification* training programme. The methodology has been formalized and has been implemented as a stand-alone CBT system.

A specific application is illustrated which involves an evaporator pilot plant. This establishes the feasibility of implementing both the integrated CBT system and the training programmes. It is shown that it is possible to use different general purpose modules to easily design and configure flexible CBT systems with advanced functionalities and to implement comprehensive CBT programmes which are easy to support and update. Furthermore, the generic structure of *Process Trainer* permits its easy adaptation to different applications. The tools and the formal methodologies are outlined for building a new functional system and defining different training programmes.

ACKNOWLEDGEMENTS

In such an occasion one is given the opportunity to express his appreciation and gratitude to the people whose contribution has been instrumental to the completion of the Ph.D. thesis. I would like to thank my supervisor, Dr. Sandro Macchietto, for his support and encouragement. While 'bringing me down to earth' and making me be specific rather than conceptual he has helped me refine and focus my ideas.

I would also like to thank Dr. Lisanne Bainbridge for her valuable contribution. Our many discussions helped me appreciate and understand the non-technological aspects of the area of training process operators. Her input has been important in meeting in this thesis issues from Ergonomics and Psychology.

I must also indicate my debt to the 'computer gurus' in the department. Dr. Costas Pantelides for helping me use SpeedUp in both an orthodox and an unorthodox way. Graham Stuart for helping me get over any problems I have had with RTPMS and for helping me address the communication issues in my system architecture. Dr. Nader Baghaei-Yazdi for helping me use effectively the computer system of the department and last but not least Richard Wood for helping me set up my various hardware configurations.

I would also like to thank Professor Roger Sargent for inciting me to undertake a Ph.D. and his subsequent support, and also Dr. Hans Michels who as the post-graduate tutor helped me persevere during a hard first period.

A special thank you goes to my colleagues and friends who have made my work and stay in London an overall rewarding experience. Although I fear I may leave someone out special thanks must go to: Luca Zullo, Yishu Nanda, Silvia Onesti, Paula Gomez, Vasilis Kouyionas, Maria Paliouri, Barry Cott, David Juarez, Ioannis Marinos, Costas Pipilis, Stavros Kokkinis, Colin Crooks, Paul Barton, Brett Westwood, Mathew Desselberger, George Yiannakides.

Last but not least, I must express my gratitude and thanks to my parents Gloria and Christofer Kassianides for always being there to motivate and support me in every possible way. I dedicate this thesis to them. My sister Vasiliki for being very supportive and understanding and contibuting to 'a home away from home' here in London. Also, Mario, Helen, Photis and Simon Kassianides for their encouragement and support. Finally, I like to thank Beatriz Marquez for all the things and plans we made. I am deeply grateful to her for putting up with me during the writing up period and for helping me persevere.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Concern for increased efficiency and safety and the increasing importance of environmental factors have in recent years forced the process industries to search for ways to improve their process operations. This led to the development of better and more efficient design strategies, parallel to progress in the development of reliable and effective process control systems. As a result problems caused by mechanical failures are said (Nedderman, 1988) to have decreased significantly in the last ten years.

This, however, has not been the case with human caused incidents. For the same ten year period, Nedderman (1988) mentions that, in the case of the Japanese nuclear industry, while mechanical failure rates have dropped from 4 to 0.6 per reactor year, problems caused by human operators have remained constant. Approximately 50% of operational events such as shut-downs in nuclear plants and most of the causes of operational failures in the chemical-process industries plants, are attributed to misoperations of the process operator or human errors (O'Shima, 1983; Sachs et al., 1986; Nedderman, 1988).

This led to an appreciation of the important role of the human process operator as well as to the consideration of how to improve the human operators' performance. This consideration has led to the development of systems and theories whose aim ranges from eliminating the operator, to providing the operator with appropriate support to perform his tasks. The most significant areas that have been pursued in order to improve the interaction of the human operator with the process are summarized in table 1.1. while in table 1.2. system categories that enable the efficient operation of a process plant and also aid the operator, are outlined. These are further examined in chapters 2 and 3 of the thesis.

Table 1.1. Concepts aiming to improve operators' performance

APPROACH	FUNCTION
Process Ergonomics	To identify the various reasons that may lead to unsatisfactory process operator performance; the nature and characteristics of human errors and the extent of the human capabilities and limitations.
Human Computer Interaction	To improve the man-machine interfaces thus providing the operator with non- cluttered displays and user-friendly environments.
Training	To improve training techniques and methods so that the operator is better prepared to perform his tasks.
Complete Plant Automation	To eliminate the human presence with respect to controlling the process on the site and provide computerized systems that will perform all the operators' tasks.

Table 1.2. Systems aimed at improving operators' performance

CATEGORY	FUNCTION
Complex Automatic Control	To take over the direct controlling task from the operator and implement complex control strategies that result in optimum plant performance.
Supervisory	To monitor the process and extrapolate the current state of the system so that future events are predicted and anticipated.
Fault Diagnosis	To enable the operator to locate failures or malfunctions in the production process through the implementation of on-line systems.
Intelligent Decision Support	To provide guide-lines and directions towards the efficient and safe operation of the plant.
Filtering Scheme	To present the operator with selected information to avoid him becoming overwhelmed.

It is noted that excluding the case of total elimination of the process operator through complete plant automation all the other approaches are aimed at enabling the operator to perform his tasks in a more efficient and effective manner. Even in the case of complete plant automation there will still be a need for expert operators who will be able to monitor the process and configure new automation modules.

The basic task of process operators is to maximize the cost effectiveness of the production process whilst minimizing any likelihood of damage to plant and personnel. Better trained operators result in safer, more economic and more efficient plant operation (Hack, 1988). Therefore, providing the necessary training, as well as improving and updating operator training practices, is an important consideration for any organization involved in process operations.

As production facilities are automated, with fewer operators and engineers overseeing the operation of wider plant areas and more sophisticated operations, it becomes increasingly important and difficult to develop and maintain the operators' ability to manage situations resulting from unexpected events, break-downs, loss of control or measurements, etc. Learning how to effectively perform their tasks takes years of practice and training, during which operators achieve both a qualitative and a quantitative understanding of the various factors that make up the industrial process. Training extends from tutoring the novice operator on the functionalities and characteristics of individual components, and on the dynamics and cause and effect relationships that define the process; to refresher training for experienced operators which covers training for fault diagnosis and fault handling.

The process industries, and in particular the power generation, oil and gas industries, have established the use of various types of Computer Based Training (CBT) systems - ranging from high fidelity training simulators to small CBT systems focusing on basic principles- to augment their conventional instructional programmes.

Training simulators have been in use for several years now in these industries, in particular in the nuclear and petrochemical sector. The simulators' ability to provide operators with 'hands-on' training in realistic environments without the complications, risks and costs arising from running the actual plant makes them particularly effective as a training medium. They are a particularly suitable vehicle for developing, maintaining and testing expertise on complex but infrequent or dangerous procedures such as commissioning, start-up, change-overs, emergency handling and shut-downs. In some industries certification procedures require operators (e.g. airline pilots) to pass proficiency tests on full scale simulators.

"At the other end of the spectrum, small computer based systems designed for self paced training have begun to enter service as basic principles and part task trainers, offering a standardised if somewhat inflexible level of training. The current generation of training devices have been developed to supplement the principal instructional process, in the main following an instructional strategy of 'drill and practice', which is suitable for learning procedures but is less satisfactory for concept learning and entirely inappropriate for debugging misconceptions" (Leitch and Horne, 1988).

1.1. Motivation for the Research

An examination of the role of the human operator in process operations indicates that in order to perform their tasks operators refer to different types of knowledge, employ a wide range of skills and perform a range of cognitive activities. Furthermore, the evolving role of the process operator leads to increasingly complex and sophisticated training requirements which in turn lead to a requirement for advanced and adaptable training methods and training delivery platforms.

Computer based training (CBT) offers these possibilities. The benefits of CBT over conventional training, in areas where CBT is appropriate, are already apparent. A summary of factors enhancing the cost-effectiveness of CBT is given in table 1.3. Table 1.4. indicates features of CBT that increase the training effectiveness. However, the benefits achieved from the use of CBT are often 'soft' and difficult to quantify in the usual accounting sense.

Table 1.3. Factors enhancing the cost-effectiveness of CBT

1. CBT applications are less costly than expensive or dangerous actual equipment
2. Effective use of CBT can allow a reduction in the number of instructional, administrative, and training support staff
3. Travel time can be reduced since instructional material can sent through communication links, thus reducing the need for costly personnel travel
4. In a flexible CBT system subject material can be rapidly updated since it is generally more difficult and expensive to change printed material than to make minor changes to the courseware

Table 1.4. CBT features that increase the training effectiveness

FEATURE	DESCRIPTION
Consistent high quality instruction capable of continuous refinement	Instructional strategies of the best instructors, as well as new elements on the basis of experience and research can be incorporated.
High quality training at remote sites	CBT courseware is easily transferable and thus, it is possible to provide good and uniform training at the particular sites without having to move instructors and/or students.
Immediate positive reinforcement and feedback	Comments are provided to the student directly and unambiguously linked to a particular action
Privacy	Students can succeed or fail in private so that the embarrassment of failure is lessened and students feel free to take more risks and explore different possibilities.
Hands-on instruction	With CBT the training process is dynamic since it depends upon the student's attention and interaction.
Idiosyncratic adaptation	A CBT system can proceed at the student's pace and provide material in a manner compatible to the student's learning characteristics.
High student motivation	The required interactive student participation and self-paced learning leads to the use of CBT systems for personal development in private time, i.e. the "open learning" concept.

For example, the saving of losses by avoiding a potentially damaging occurrence, such as a runaway reaction, due to the skilful handling of a previously rehearsed situation is unlikely to appear as such on any bottom line. Benefits are none the less real and often quite significant (Gran et al., 1988; Elston and Potter, 1989).

Although a great range of CBT systems has been developed, their routine use by the process industries especially with respect to training simulators is presently limited. There are several factors which have inhibited the widespread use of CBT for the training of process operators. The greatest being the effort and costs (presently extremely high) involved in building, supporting and updating CBT systems and courseware. As a result such systems generally tend to be case or problem specific implementations, often deeply embedded in bespoke hardware and software. Therefore, subsequent maintenance and changes to reflect, for example, plant or procedure modifications, new training objectives etc. become often difficult to implement leading to a rapid obsolescence of the entire system.

This is especially true in the case of training simulators whose current costs can be very high (10^5-10^6 US$) and must be incurred up front. As a consequence the routine use of training simulators is still limited to high risk, highly regulated, or high capital situations such as nuclear plants, oil refineries, off-shore processing on platforms etc. where a substantial investment in the area has been observed in the last decade.

Particular limitations arise in the implementation of training simulators at a justifiable cost due to the lack of readily available technology and expertise. Major difficulties can be identified in the initial development of models for plant and control systems, robust and fast numerical techniques for the solution of the model equations in real-time, operational and training sequences, and sophisticated user interfaces. Furthermore, a major proportion of the operating costs in simulator training results from the need for expert instructors.

This last factor further emphasizes the need for stand-alone computer based training systems especially ones that combine qualitative and quantitative domain representations in order to provide training programmes which integrate theory and practice. It is currently difficult to integrate these two approaches to training in particular to merge dynamic simulation with tutorial methods. Very little is currently available in this respect, and this is an active research area.

Another area of interest involves the formal definition of training programmes for CBT implementations. While a range of training programmes have been described, very little has been done to define formal implementations as CBT systems. A particular case is in the area of training for fault diagnosis. While an active research area, little has been done in formalizing a training programme that may be implemented as a stand-alone CBT system.

Finally, while a range of CBT authoring systems abound, there is none available with an emphasis on the process industries and its particular requirements for the use of advanced simulation capabilities. A flexible system that overcomes the limitations outlined in the previous paragraphs and also provides the medium for easy development and delivery of training both in basic principles and in 'hands-on' plant operation whilst minimizing the involvement of instructors could lead to substantial cost reductions while increasing training effectiveness.

Referring to a recent market forecast for trainers and simulators in the Process and Power Generation industries, Leitch and Horne (1988) indicate that the total annual European demand was estimated at £118m with an associated estimated growth rate of 16%. They further indicate that it is unlikely that supply could meet this demand using the technology currently in use and that the introduction of more effective design methodologies and part solutions is clearly necessary to meet this demand.

1.2. Thesis Objectives

The principal objective of this thesis is to propose the design of a system architecture that provides tools which allow the easy development and implementation of flexible and easy to maintain CBT systems and courseware for both basic principles and 'hands-on' plant operation whilst minimizing the involvement of instructors. A second objective is to propose methodologies which utilize the tools offered by the system to produce comprehensive CBT programmes for a range of training purposes in the process industries. The final objective is to develop a Training Demonstrator that utilizes the tools and methodologies in a variety of ways for a specific application which involves an evaporator pilot plant. The underlying goals of the three main objectives can be summarized as follows:

1. To meet the process industries' requirements and demands as they relate to the training of process operators.
2. To propose a novel design which allows the implementation of CBT systems and courseware for a range of training purposes in the process industries.
3. To formalize the definition of training programmes for process operators in view of their implementation as CBT.

In order to achieve these goals the following steps have to be taken:

1. To review requirements for operator training in the process industries.

   It is important to examine the role and characteristics of the process operator as these will indicate the broad training issues that need to be addressed. Considerable work has already been done in this respect. However, it is necessary to bring together the findings of the previous work, with the process industries' current and anticipated requirements and demands as they relate to the training of process operators.

2. To review the state of the art in operator training technology.

   Currently many different techniques are being applied independently for the training of process operators. There is however scope for integrating various techniques for greater effectiveness. In particular considering the general area of CBT (not only as it has been applied to operators) should indicate techniques and experiences that may be utilized to formulate a functional specification for an effective CBT system.

3. To produce a list of required functionalities for CBT systems.

   The findings of the two reviews outlined above should indicate the required functionalities for CBT systems for process operators. Currently such guide-lines are not available.

4. To explore the feasibility of using existing general purpose building blocks for the design and implementation of training systems.

   The current state of technology as it relates to the availability of existing engineering orientated computer software for modeling, simulation, control, operations management, and expert system development and the decreasing costs of computer hardware offers the possibility for using available platforms for the development and implementation of CBT systems for process operators.

5. To propose a system architecture design that allows the development and implementation of CBT systems of advanced functionalities.

   Take up the challenge of using the building blocks mentioned above to design a system architecture that will serve as a platform for the development and implementation of CBT systems with an emphasis on the process industries and its particular requirements for the use of advanced simulation capabilities. This should thus meet both the technological and operator training challenges identified by the two reviews.

6. To implement the proposed design.

   The implementation of the proposed design should establish its feasibility and define the scope for developing and implementing different training programmes.

7. To propose a training methodology which utilizes the above tools to produce a comprehensive CBT programme.

   Having implemented the proposed design the need arises for a methodology which makes effective use of the training system development platform in order to address different training objectives.

8. To formally define training programmes for their implementation as CBT systems.

   To date technological limitations particularly with respect to the development and implementation of dynamic simulation models and their solution in real-time, and the easy definition of flexible qualitative CBT courseware, made the integration of the two techniques extremely difficult. As a result CBT has been used primarily to augment traditional instructional methods. However, the capability to use the tools offered by the implemented system should allow the easy and effective integration of the two approaches and effectively merge dynamic simulation with tutorial methods. Novel training programmes could be defined to exploit these added functionalities and be implemented as CBT.

9. To develop a prototype Training Demonstrator.

   This should demonstrate the effectiveness of the proposed techniques and methods.

1.3. Thesis Outline

Following this brief introduction to the thesis, chapter 2 examines the characteristics of the process operator. An attempt is made both to identify the nature of the tasks that the operator has to perform as well as the most likely causes of human error and their characteristics. The examination is performed in view of the contribution that improved training may have on the operators' performance. Chapter 2 also examines various approaches that have been followed to improve the efficiency of process operations and examines their effect on operator's tasks.

Chapter 3 includes an overview of training issues and a detailed survey of computer based training (CBT) as they relate to process operator training. A methodological approach to developing training programmes is presented. This includes: identifying the trainee characteristics, performing a training needs analysis, defining the training objectives, performing a task analysis and finally selecting the most appropriate training method. The survey first presents a cost/benefit analysis of CBT. Then it examines each of the branches of CBT (Computer Managed

Instruction, Computer Assisted Instruction, Intelligent Computer Assisted Instruction and Training Simulators) providing an overview of both the theory and relevant applications.

In chapter 4 an integrated architecture for computer based training systems is defined which allows great flexibility in the development and implementation of advanced and sophisticated CBT systems. A functional definition of each of the components of the architecture is given and also several training system configurations that have been achieved are presented.

Chapter 5 presents *Process Trainer: A Tutoring System for Operations Staff.* This chapter examines three training programmes each addressing a different process operation training need and presents a formal methodology for using the tools provided by the system architecture to address different training objectives.

Then in chapter 6 a functional definition of *Process Trainer* is provided. This involves the description of a set of general rules that are used to define its structure, the different ways that a specific domain may be represented within it and finally the general knowledge base structure of each of its components.

Process Trainer and the complete system architecture were applied to a particular case study involving an evaporator pilot plant in the department of Chemical Engineering at Imperial College. This illustrative application is presented in chapter 7.

A discussion of the various issues raised by this thesis is given in chapter 8. There the final conclusions of this research are drawn and its significance is indicated. Finally areas requiring further work are presented in chapter 9.

Chapter 2

THE HUMAN OPERATOR IN PROCESS CONTROL

The first step to training is to understand the range of tasks which the trainee will have to perform and to identify the nature and the most likely causes of human errors that may occur. The process operators tasks and the most commonly observed errors, as well as various approaches aimed at improving process operations are examined in this chapter.

2.1. Operator Tasks

The main objective of process control operators is to maximize the cost effectiveness of the production process whilst minimizing any likelihood of damage to plant and personnel. The abilities that the operator needs to have in order to achieve this objective, encompass the whole range of human abilities from perceptual motor tasks, such as tracking and manual control, to cognitive tasks, such as decision making and problem solving. In particular the operators' process control subtasks (summarized in table 2.1.) as described by Eberts (1985) are: monitor, control, interpret, plan, and diagnose.

With the spread of automation through sophisticated computer control and on-line decision support systems, and the increasing use of large, complex production systems, process control is increasingly becoming a cognitive activity for the operator rather than a perceptual and control task. Thus, there has been a decline in the demand for manual control skills and a consequent increase in the use of what can be called *mental* skills. "These skills involve decision-making processes and other intellectual activities which are as yet little understood" (Bainbridge et al., 1974).

Table 2.1. The human operators' process control subtasks.

SUBTASK	DESCRIPTION
Monitor	Monitor for seldom occurring events (vigilance, signal detection, separation of signal from noise).
Control	Keep system on optimal course (tracking).
Interpret	Separate random fluctuations of system from actual course; filter out noise (categorisation, estimation, filtering and quantization).
Plan	Set goals and strategies; use resources efficiently; sequence tasks; use heuristics; develop strategies (decisions, resource sharing, allocate resources).
Diagnose	Identify the problem when a fault occurs.

As automation proceeds the quality and nature of the process operators tasks is continuously upgraded. Future "operators will be called on to monitor and supervise the process operation at large rather than the level of control loops; to interact with the control computer for advice, to provide preferences, to check the computer's logic, and to intervene to safeguard smooth and reliable operation in cases of precipitous change" (Stephanopoulos, 1987). To date, the role of the process operator has undergone three levels of 'evolution' (Brouwers, 1984). These three levels are: Intensive Monitoring, Active Monitoring, and Survey Monitoring.

Intensive monitoring is mainly associated with manual control (see figure 2.1.). The operator forms part of the closed loop of the control system. He performs actions in order to minimize deviations between output and desired values. Therefore, the operator observes the displays almost permanently since the information is needed for the control of the system.

Active monitoring is associated with automatic control (see figure 2.2.). Here both the operator and the automatic controller execute actions in order to minimize deviations between output and desired values. Monitoring activities under this configuration consist of regular inspections of the state of the process. Sometimes the operator may take the initiative and take control actions instigated by the display-inspection.

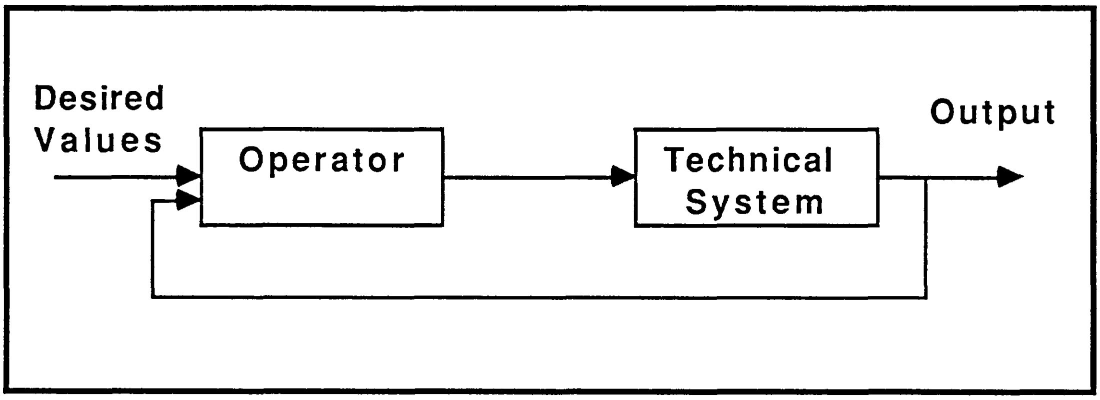


Figure 2.1. Schematic representation of Manual Control

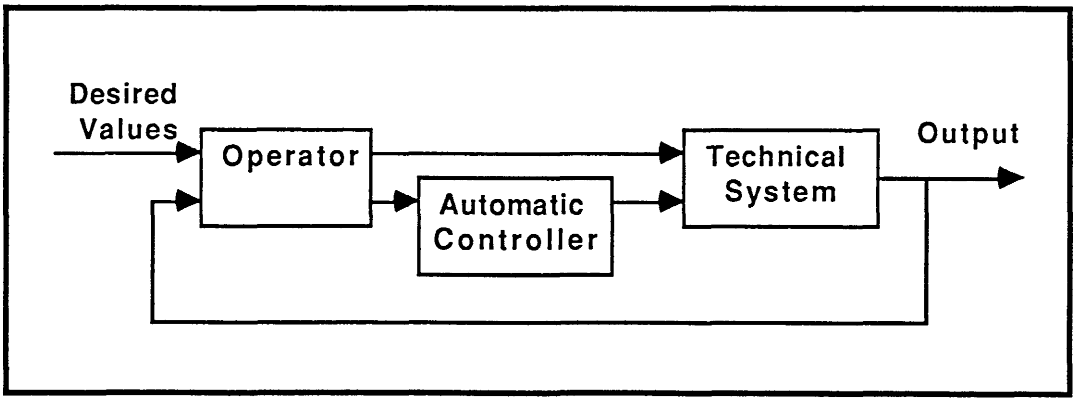


Figure 2.2. Schematic representation of Automatic Control

Survey monitoring is associated with supervisory control (see figure 2.3.). In this configuration the process is controlled by the automatic controller and the operator is no longer part of the control loop. His task here is mainly a monitoring one. The operator can only interfere by setpoint-control and control activities seldom follow the monitoring actions.

Even though the active participation of the operator in process operation has diminished with advances in technology the nature of his subtasks as given in table 2.1. has not changed. Rather the frequency of active participation has decreased, thus, raising issues such as alertness and need for thorough preparation so that the operator can take over when necessary and save a situation.

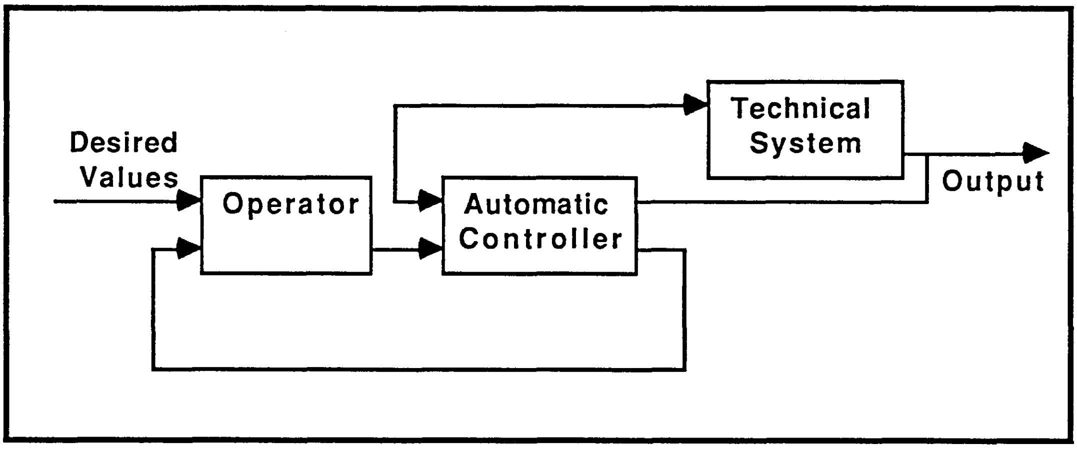


Figure 2.3. Schematic representation of Supervisory Control

With even further technological advances the role of the operators' tasks is expected to take a new dimension. *Exception monitoring* will be the result of extended or extra supervisory control where all action is left to the computerized control system and the operator's intervention is requested by the system whenever that is necessary. "At such moments the systems provide relevant information and the operator can request additional information that will enable him to perform what is required of him" (Brouwers, 1984).

2.2. The Expert Process Operator

An important consideration is the identification of the various characteristics that describe the expert process control operator. The underlying objective of this analysis is to enable the transfer of expert knowledge to novice operators through appropriate training methods. Thus far, two main qualities have been attributed to the expert operator.

First, the expert operator is very good at knowing what to expect from the operation of the system. This knowledge can then be used to effectively achieve and maintain different operational states, produce desired changes in production, and identify deviations from normal operation. This is particularly apparent in the latter case. In such situations an expert operator's familiarity with the system enables the efficient location of 'problem sources' (for example faults and/or malfunctions). Thereafter, his extensive knowledge of the cause and effect relationships that define plant operations enable him to effectively handle most fault management situations.

Second, expert process control operators have available to them a spatial representation of the system which enables them to perform mental simulations of the plant operation. In this way they can predict future states of the process, anticipate the effects of control actions and also efficiently identify the point where a malfunction has occurred.

People appear to work from an 'internal' model of the system they are controlling (Eberts, 1985). Very little is known about the form of this internal model, but studies (for example: Bainbridge et al., 1974) on the subject's running memory for variables in the system show that they keep mental track of important variables, even though these may be displayed in front of them.

2.3. Common Operator Errors

Having examined the process operators' tasks and the evolution in terms of the role played by them in process operations it can be concluded that since the level of problems due to human error has remained significant, either the very nature of the causes of human error is changing or that the sources of human error have not been adequately identified or that they have not been dealt with efficiently.

People are often the source of operational errors, leading, at best, to inefficient plant operation, at worse, to disastrous events. Operator errors have been divided into three categories (Black, 1987) by psychologists: (a) errors in perceiving events around them; (b) errors in making decisions; and (c) errors in executing decisions. Such errors result from lack of specialized knowledge, low-level skills or experience, limited information on the state of the process, excessive amounts of confusing process information, or reduced physical or mental competence (Stephanopoulos, 1987; Kletz, 1988).

The flexibility provided by current generation computers means that it is possible to present to the operators all the plant information available, and this is what usually happens. The operators, therefore, have to handle more information, are less familiar with any part of it, and have to take more complex decisions. The amounts of information involved can be considerable.

As a result, when things start going wrong in the plant, operators are often overwhelmed by the amount of information being produced, become stressed and start making mistakes. As pointed out above these mistakes include wrongly perceiving the plant status, wrongly diagnosing problems, performing wrong corrective actions and responding to problems in the wrong order. This, in turn, causes greater stress leading to even more mistakes. This phenomenon is known as *cognitive overload* (Sachs et al. '86).

In reporting on accidents in Petrochemical Complexes in Japan in 1981, O'Shima (1983) notes that approximately 60% of all incidents were caused from human error. Trevor Kletz (1985) states that "accident statistics from many companies show that over 50%, sometimes as many as 90%, of industrial accidents are due to human failing."

Furthermore, by closely considering a case study of a particular Japanese Petrochemical Complex for which a detailed analysis of equipment and operational failures is given (reproduced in tables 2.2. and 2.3.) clear examples may be identified of areas where the performance of the operators could be improved and, hence, increase the cost effectiveness of process operation.

Table 2.2. Case study of equipment failures in a Japanese petrochemical complex (O'Shima, 1983).

<u>EQUIPMENT FAILURES</u>	317
a) Defects in structural design	102
b) Improper materials	38
c) Defect in fabrication	52
d) Instrumentation defect	11
e) Material degradation	68
f) External load, impulse	11
g) Others	31

In another report dealing with the causes of ammonia plant shut-downs world-wide, Williams and Hoehing (1983) maintain that human-error-caused shut-downs occurred every 3.5 years. Their findings show that the percentage of the number of shut-downs attributed to human error ranged from 4% for large-tonnage ammonia plants to approximately 3% for reciprocating ammonia plants. Furthermore, the percentage with respect to the number of down-time days was approximately 1% and 9% respectively.

These results though indicating the presence of human error may underplay the importance of the human factor in process operations. It is, however, noteworthy to point out that while a lot of human errors may not lead to major plant shut-downs, they may still lead to considerable accidents that may involve considerable damage to equipment or personnel. Examining the various human errors one may find that there are a number of alternatives that could lead to their minimisation.

Table 2.3. Case study of operational failures in a Japanese petrochemical complex (O'Shima, 1983).

	OPERATIONAL FAILURES	482
a)	**Incomplete communication**	**14**
	- Wrong information	
	- Misunderstanding based on ambiguous information	
	- Improper means of communication or wrong place of signs	
b)	**Malrecognition**	**48**
	- Inaccurate recognition	
	- Operation under a false impression	
c)	**Misoperation**	**48**
	- Unconsciously executed operation according to custom	
	- Handled by reflex action without consideration of danger	
	- Manipulating wrong identical devices	
	- Misoperated in spite of having enough knowledge	
	- Not aware of mistakes due to lack of feedback	
d)	**Misunderstanding**	**39**
	- Did not know operational procedures to avoid urgent conditions	
	- Forgot operating procedures	
	- Felt safe based on experience	
	- Hesitated to make a decision owing to complicated conditions	
	- Omitted to make a decision owing to urgent conditions	
	- Paid attention to other requirements	
e)	**Poor skill**	**20**
f)	**Imperfection of standard operating procedures**	**112**
g)	**Imperfection of instructions**	**46**
h)	**Inspection failure**	**92**
i)	**Maintenance failure**	**50**
j)	**Technical unknown factor**	**16**
k)	**Others**	**29**

Errors occurring due to lack of specialized knowledge -for example a side reaction that may take place and lead a reactor to a dangerous situation under certain conditions- could be overcome by means of appropriate training, or through the implementation of a supervisory system, or even an intelligent decision support system. Where training would aim to instil the knowledge and formal problem solving methodologies to the operator the other two methods could check the status of the process and inform the operator or even guide him through the appropriate course of actions to avoid the incident.

Dealing with *errors occurring due to lack of skill and experience* could be overcome by the use of simulators for training. Since most chemical plants are highly automated there is little chance of accumulating experience on-the-job through direct interaction. Fault diagnosis systems, supervisory systems or intelligent decision support systems could pin-point the problem or suggest corrective actions. However, they may have an adverse effect on the human operators' long term effectiveness. While they may help the operator to deal with a particular incident the educational value of this experience is negligible. The fact that he may have dealt with the incident effectively, does not imply that he has understood the underlying reasons for the incident and that in the future he will successfully cope with the same or a similar incident if the on-line system is not available.

Errors due to excess or lack of information can be dealt with by using filtering scheme systems that select the amount of information presented, intelligent decision support systems that guide the operators' actions and through the provision of appropriate training. Through the understanding of the cause and effect relationships in the plant and with an extensive familiarisation of the human computer interface the operator should be able to focus attention on important events. Following an extensive 'theoretical' training session, and a session in an appropriate training simulator environment where unexpected events can be played out, could provide the operator with the necessary experience. This would allow the operator to be trained in similar circumstances and enable the development of a methodology that will help him to focus on important information.

Errors due to physical or mental fatigue are inherently difficult since they have to do more with the structure of the shifts and the accumulation of responsibilities which the operator has to meet. Effective selection of presented data, reasonable workloads and effective training programs that fully prepare operators in view of their tasks, should improve human performance with respect to such errors.

2.4. Improving Process Operations

The need to increase efficiency in process operations led to the development and application of various concepts and systems. These were briefly summarized in tables 1.1. and 1.2. respectively. In general the approaches aimed at improving the efficiency in process operations can be divided into two distinct categories. In the first one the operator no longer constitutes an active part of the process while in the second one the operator is still an integral part of the process.

The first category resulted from the belief that plant safety was guarantied by a good design, and that the operators' contribution was generally negative. Therefore, alternatives to human operators were sought. One such alternative replaces the man by simulating him on a controlling system computer, which has the advantage that it can work accurately and without fatigue for indefinite periods of time. This led to the development of various on-line computer systems that aimed at fully automated plants.

The second category arose partly due to the present needs in the process industries until full plant automation is achieved; partly due to the technological breakthroughs achieved in view of complete plant automation; and finally from the understanding that even in fully automated plants there will still be a need for human involvement even if it is implicit. Thus methods and tools that aim to improve the process operators' efficiency are currently being developed and applied.

2.4.1. Complete Plant Automation.

The development of complex computerized control strategies and effective operator aids (diagnostic, supervisory, decision support, operations management e.t.c.) promise that in the future complete plant automation could become a reality.

However, the question of human limitations takes a new dimension under the perspective of automation. As the role of the operator increasingly becomes supervisory *(monitoring by exception)* familiarity with the process and the system will weaken. Therefore, the effectiveness of operators' to deal with emergency or other situations where direct intervention is required, is questioned.

Studies suggest (Bainbridge, 1974) that the internal model that operators develop for the system can only be kept in mind and updated regularly if they are more or less continuously involved with on-going control activity. A recent survey showed that "operators are highly reluctant to see their ability to control the system eroded, by loss of initiative to the computer system" (Alty et al. 1986). Over-dependency upon automation systems may have negative effects since in cases where they are not available, operators may not be able to perform their tasks independently.

As a result it is seen that in the case of complete plant automation, even though the operator no longer constitutes an active part of the process, the human presence still remains. However, the nature of the job ceases from being a continuous interactive process but becomes one of immediate intervention. Therefore, future *'operators'* will have to be experts in the process and the plant and have the knowledge and the ability to handle unlikely situations. Consequently with the move towards automation the requirements with regard to professional training do increase. This emphasizes the need for more comprehensive and sophisticated training that will be administered by increasingly complex training systems.

2.4.2. Systems Aimed at Improving Process Operation

A number of systems have been devised which aim to increase the efficiency and safety of processes. While these constitute means of enhancing the effectiveness of the process operator who is an integral part of the process, they also form the backbone of the drive for complete automation. They include among others: (a) advanced control systems, (b) supervisory systems, (c) fault diagnosis systems, (d) intelligent decision support systems, and (e) filtering systems.

2.4.2.1. Advanced Computerized Control Systems

To enable the operators to perform their tasks with higher efficiency, complex control and information systems have been developed that take over continuous and interactive control from the operator and implement complex control strategies that result in optimum plant performance. The function of these complex control and instrumentation systems is threefold (Zwaga and Veldkamp, 1984):

1. To reduce the need for operator intervention and to automatically control the plant.
2. To provide up-to-date information about the process plant and enable optimum control strategies to be implemented by the operators
3. To reduce the risk of damage to plant and personnel by automatically shutting down part or all of the plant when process variables move out of their normal operating ranges.

The implementation of these advanced computerized control systems have significantly increased the efficiency of process operations. However, they are not fool proof. A wide variety of problems can occur (Sachs et al., 1986) which will cause the control system not to fulfil these functions including: valves sticking, transmitters failing or sticking, transducer circuits going open or closed circuit, and operators making errors such as forgetting to operate shut off valves or wrongly adjusting control loop set-points.

Any of these problems can cause the plant to run outside its operating range and so increase the risk of damage to plant or personnel. This is the point where an operator's performance becomes very important. The operator will need to take corrective action sometimes without the help of the control system or any support system and, therefore, he needs to be well trained to be able to cope with unexpected situations.

2.4.4.2. Supervisory Systems

Supervisory systems are based on a dynamic extrapolation of the current state of the process so that future states are predicted and can be examined. These are based on a dynamic simulation of the process which enables the study of the dynamic profiles of the various variables which in turn indicate where the process is leading and thus can predict undesirable situations. The advantages of such systems serving as predictor algorithms could be great. However, high fidelity dynamic simulations of the process need to be implemented to avoid misleading results.

2.4.4.3. Intelligent Operator Aids

Intelligent operator aids include fault diagnosis systems, intelligent decision support systems and filtering systems and can thus be examined as a group. Advances in artificial intelligence (AI) have prompted researchers to investigate the potential for using AI techniques, particularly expert systems, for the development of operator aids. This led to the belief that in the near future the process of diagnosis and the taking of the appropriate control actions will be done automatically.

However, experience has proved that presently "it is impossible to foresee all the abnormal situations that might happen in the plant, thus to prepare the comprehensive system of counter measure to plant failures" (O'Shima, 1983). More advanced systems will have to be developed that will meet these demands. In the meantime emphasis has been placed on developing systems that will aid the human operator perform his tasks in a more efficient and effective manner.

AI systems provide sophisticated problem solving techniques to assist operators of complex systems to identify and evaluate problems and their potential solutions. This is particularly useful in cases where the number of alternatives is beyond the usual scope of operator experience.

Expert systems can also serve to preserve expert knowledge in an industrial plant. Their knowledge-bases can be expanded and/or modified so that they will encapsulate further obtained knowledge. Moore and Kramer (1985), observe that "an expert system, once the knowledge is defined, could serve as a training vehicle for the new operators. In many cases expertise is built-up over many years of operation, and is lost when an employee leaves or retires. The knowledge base of an expert system can be expected to become a corporate resource, available without disruption for training, simulation and other purposes as well as being used for advice and control for plant operations."

Consequently, research is currently under way to define and design man-machine interfaces that based on expert systems will:

- provide lists of abnormal operating conditions;
- identify possible causes of failure and screen for the most likely ones;
- demonstrate the effects of occurring or anticipated faults and provide a code of standard operational procedures;
- provide sensible monitoring of standard operational procedures by periodically checking the plant's operational status;
- chart operational trends and extract dominant operational features, which are provided to the technician with explanations and suggestions about the future course;
- provide graphs of the process with zooming capabilities and multiple levels of abstraction from very detailed to coarse outlines;
- provide graphs and tables of process performance per unit time (differential) over a period of time (integral), as well as detailed process accounting using internal pricing for the various streams.

2.5. Summary

In this chapter the human contribution to the safe and efficient operation of a processing plant was examined, by focusing on: (i) the process control operator's tasks, (ii) the common operator errors, and (iii) the various approaches employed which aim to improve the operator's performance. This examination pointed out a number of factors that need to be considered and which have a direct implication in the development and implementation of effective training programmes for operators. In summary, the following comments can be made:

1. Operators employ a wide range of skills in order to perform their tasks successfully.
2. Operators employ 'internal' models of the process in order to perform their tasks.
3. Human error is responsible for a great proportion of industrial accidents.
4. Effective training could lead to a minimisation of human errors.
5. As automation proceeds the role of the operator evolves increasingly towards a supervisory one.
6. Even with complete plant automation an implicit need for a human operator remains.
7. As the role of the operator evolves with automation so does the sophistication and complexity of the training requirements.
8. With the increasing use of support systems for process operators there is a growing need for training to maintain the operators' abilities.
9. Training programmes for operators must address a variety of needs ranging from background theory and 'hands-on' training in plant operation, to decision making and problem solving methodologies.
10. There is a need for diverse and sophisticated training delivery platforms that can accommodate the range of training needs.
11. Training programmes need to be flexible and easily modifiable to accommodate the changing role of the process operator.

The examination of the process operator's contribution to plant operation has led to the identification of the need and the broad requirements for appropriate training programmes. In the next chapter the general area of training and how it is applied in the case of the process operator is examined in detail.

Chapter 3

TRAINING THE PROCESS OPERATOR

In this chapter the underlying issues which form the planning stages of training design and which define the effectiveness of a training programme shall be briefly examined, along with the various training methods that are currently in use for the training of process operators. Then, a detailed survey of computer based training will be presented which identifies the area's current state of the art and limitations, as well as possible areas of improvement. These will then form the basis for the development of the integrated system architecture and the development and implementation of the training programmes that are described in chapters 4 and 5 respectively.

In general the methodology followed for training design involves the following steps. First, it is important to understand the characteristics and capabilities of the people to be trained. Then one has to identify the training needs, define the training objectives, and perform a task analysis. Following this formal definition of the domain and of the training process, the trainer needs to consider alternative training options and then select the appropriate training method.

3.1. Trainee Characteristics

The identification of the range of knowledge, level of expertise, motivation and learning style of the individual and/or the group of individuals to be trained is very important. While it is necessary to specify the objectives of a training course in terms of the capabilities that the trainees are expected to acquire by the end, it is equally useful to specify their capabilities at the start. This allows the design of an effective training course which suits the training needs, does not include much redundant material, and utilizes the most appropriate instructional strategy and delivery method (Dean and Whitlock, 1988; Wetherill and Wallsgrove, 1986).

The range of prerequisite knowledge, skills and capabilities at each level of competence is best indicated by reference to different groups (eg novice v. expert) employed to perform the same tasks.

3.2. Training Needs Analysis

The main function of a training needs analysis is to identify deficient performance, establish its cause and to propose an appropriate solution. There are generally two procedures of training needs analysis (Dean and Whitlock, 1988). The first procedure, *reactive analysis,* is employed to analyse problems of inadequate performance on existing tasks. The second procedure, *proactive analysis* is faced with an impending performance problem as a result of the introduction of new systems or operational procedures in which case few, if any, of the existing staff possess any relevant knowledge or skill . *Proactive analysis* is also applied for the formulation of the training needs of novice trainees.

"Careful analysis of training needs is essential to ensure that the tasks are described sufficiently precisely for specifying training and that the correct training options are selected" (Shepherd, 1989). It is important that the training needs are identified early in the design cycle so that problems that could be solved with non-training options (such as (Shepherd, 1986): modifying displays and controls, providing the operator with job-aids; and re-allocating functions between operators, plant and equipment) are identified and dealt with. This will allow the proper preparation for training design.

3.3. Training Objectives

Training objectives define the knowledge, skills and attitudes that the trainee will attain as a result of the instruction. These in turn are defined by the set of tasks which comprise them. Thus, the training objective for a course or session is a summary of the task analysis and the associated level of proficiency in carrying them out. It defines the task in terms of the competence to be learned.

Two major types of objectives have been identified (Lysaught and Williams, 1963). The first type are *immediate objectives*. These are usually stated as something concrete to be learned, or as some specific knowledge, understanding or skill to be acquired. The second type are *ultimate objectives,* and relate more to the long-term development of the learner and the systematic use of the subject matter .

As far as possible, all the objectives should be defined in operational, observable, and measurable terms in order to facilitate the training design and its subsequent evaluation. It is often helpful to state clearly the objectives of any piece of instruction (what the trainee is expected to learn in order to carry out the part of the task in question) since it will allow the trainee to place emphasis upon key features and recognize when the learning objective has been attained. This will also enable the instructor to focus attention on the key issues to be taught and recognize when the trainee is able to demonstrate competence (Lysaught and Williams, 1963; O'Shea and Self, 1988; Dean and Whitlock, 1988; Shepherd, 1989).

For example in the case of process control operators one *ultimate objective* may be to 'perform a plant start-up from cold'. This is a *life-long objective* (Dean and Whitlock, 1988) as it will be continuously refined and improved according to experience and new technological advances. On the other hand an *immediate objective* that may form part of the stated *ultimate objective* could be 'set up the liquid circulation'.

3.4. Task Analysis

As stated in the previous section training objectives are defined by the set of tasks which comprise them. Therefore, the identification of these tasks is an important aspect of any training design. Task analysis aims to identify precisely what the individual trainee needs to know. It is the "analysis of a learning task in terms of an ordered set of required capabilities" (O'Shea and Self, 1988).

A task is a logically related set of actions required for the completion of a job. Tasks and activities may be broadly classified (Holding, 1987) as either *verbal* or *motor*, or as composite of both of these. The verbal category includes linguistic activities and also cognitive processes such as judgement, rule learning, and problem-solving. The motor, or perceptual-motor, category includes all of the other tasks in which bodily action plays a part. Tasks may be *simple* or *complex*. There is also a distinction between *discrete* tasks, such as setting a setpoint, and *continuous* tasks, such as maintaining steady state. The discrete tasks may in turn be either *single*, or else *serial*, as in the repeated procedures used in many industrial tasks.

In particular task analysis for instructional purposes needs to describe both the expert's performance and also the performance characteristics of novices in an attempt to discover or point key differences between them, thus suggesting ways of arranging experiences that will help novices become experts. "Instructional task analysis should elucidate the relations between activity during learning and competence that results from learning. It should suggest ways of organizing knowledge to assist in acquisition, recognizing that this organization may differ from organizations that are most efficient for expert use of that knowledge" (Resnick, 1976).

A number of approaches have been developed for task analysis. Thorough reviews can be found in Duncan (1974), Resnick (1976), and Gregg (1976).

Shepherd (1985, 1986) describes and gives examples of an approach "particularly suited to the analysis of process control jobs".

"*Hierarchical task analysis* commences by describing the job or task in terms of an operation -an instruction to achieve a goal, such as 'run plant', 'operate desalination plant' or 'operate compressor'. The analyst then re-describes the operation in terms of a set of sub-ordinate operations and a plan which governs the conditions under which each of the sub-ordinate operations are carried out. Any operation is therefore equivalent to the set of its sub-ordinate operations as carried out according to their plan. Each of the sub-ordinate operations can be re-described further if the analyst requires, again in terms of sub-ordinate operations and a plan" (Shepherd, 1986).

3.5. Training Methods

The nature of the task to be learned will determine the content of the training programme as well as the appropriate training method. An abundance of instructional methods have been employed to deliver training. It is therefore, important to identify which method(s) is best for achieving the training objectives. Alternatives include (Wetherill and Wallsgrove, 1986) lecture programs, slide programs, videotape or movie presentations, home study, workshops, on-the-job training, simulation training, computer based training or some combination of these.

There are no clear cut rules as to which method is best for any particular situation. Often a combination of methods will prove to be more effective than any individual one. The selection of the most appropriate method, or combination of methods, for specific objectives is necessary if a structured and effective training programme is to be achieved. Nevertheless, more often than not the final selection will be determined by time and cost constraints.

3.5.1. Lectures

The traditional approach to training process operators has generally been training through lectures and on-the-job. Lectures are useful especially in presenting theory and general issues where no individual attention is required. This could involve "teaching basic knowledge about the plant, for example its structure and function, and about elementary physics and chemistry, is often a useful way of introducing the trainee to the plant and process. It can help in teaching the names of parts of plant and equipment, justifying certain procedures and safety measures and it can often be quite motivating if done well" (Shepherd, 1986).

The application of various training tools such as slides, video presentations have enhanced the transfer of knowledge through lectures. Furthermore, lectures are cost effective since one trainer can at the same time teach a number of students. In addition to that, the trainer has a first hand understanding of the effectiveness of the training session. Therefore, he can make necessary on the spot changes. For example, he could provide the trainees with various examples in an attempt to clarify and explain difficult concepts and thus ease the learning process.

However, lectures have their limitations. They are remote from the real world, and not well suited to training people that have to perform practical tasks, as is the case of process operators. Nevertheless, the initial theory could be presented through lectures. However, since all trainees are expected to perform their job up to a certain standard, limitations arising from the lack of personal attention and individual training may lead to superficiality and misconceptions.

3.5.2. Self-Study

Self study is based on the assumption that the trainees are self motivated and that they can follow courses at their own pace. It is not aimed at trainees that are being trained to perform certain practical tasks.

An example of this approach to the problem of training operators is the *Open Learning Course for Process Operators* (McCaughey and Geary, 1988), a long term scheme that aims to give the process operator a better all around education. It is a method that has been receiving considerable attention lately and its main aim is to meet specifically the needs of shift personnel. As Geary (1988) points out, open learning "is a student oriented activity based on carefully-written and self-explanatory material with tutorial back-up and reassurance, balanced workloads and sufficient incentives to satisfy self motivated students."

The advantages of a successful self study program are that the trainee has more or less total control on the training process as he can organize his time and proceed with the course at his own pace. Furthermore, self study is cost-effective as it is carried out on the students time and therefore it does not interfere with actual shift work, Additionally, once the training material has been prepared by the trainer they can be applied time and time again.

The main disadvantages of self-study are that it is entirely dependent upon student motivation and that it is best suitable to theoretical training rather than training for practical tasks and real environment operation.

3.5.3. Workshops

Workshops provide the opportunity for trainees to assemble and discuss the subject matter and exchange views and experiences. It is an effective method since it can pin-point issues that have not been clearly understood by the trainees in lectures or during home study. In the case of refresher courses for experienced operators, workshops are a valuable means of discussing plant operation as has been observed in real life. However, the workshop method assumes that some other form of training has preceded and so is an additional method that enhances a "trained" operator's understanding.

3.5.4. On-the-Job Training

On-the-job training is an effective way of providing the trainee with a realistic understanding and a good first hand feel for the tasks that he will have to perform. "It also provides an opportunity to help integrate the new trainee into the working environment" (Shepherd, 1989). During on-the-job training the trainee is usually teamed up with an expert operator who acts as the instructor.

The expert operator points out things that are system and plant specific and tries to provide the novice operator with the specialized knowledge that he (the expert) has attained through operating the plant in the past. "This requires the instructor to describe and demonstrate various parts of the task to the trainee, then observe and correct the trainee as he attempts the task for himself" (Shepherd, 1986).

However, on-the-job training has its limitations. It is at best an inefficient training procedure and can lead to the perpetuation of poor working strategies. The fact that the trainer is an expert operator does not guarantee his ability or willingness to teach or even communicate so as to transfer his knowledge to the novice operator.

Another important consideration arises from the highly advanced control systems and 'efficient' designs that exist in industry. It is very possible that a new operator will not be exposed to an alarming situation during training and even if a critical situation arises, it is unlikely that the novice will benefit from it. First, the expert operator will be more concerned with dealing with the problem rather than teaching. Second, the probability that the next critical situation that arises will be the same is very low. Therefore, this experience may not prove useful.

For the same reasons it is possible that a new operator will not experience start-up or shut-down during training since, nowadays, there may be a time period of a few years before a plant will have to be shut-down. As a result this form of training needs to be supplemented by other methods.

As a result on-the-job training is best at bridging the gap between theory and practice in tailored environments, and the real world. However, it lacks in the areas of efficient knowledge transfer and adequate preparation for general problem solving methods.

3.5.5. Simulation Training

"Simulation training means using some kind of representation of the task to be learned to provide a trainee with the opportunity to practice aspects of operating skill safely within a training programme" (Shepherd, 1986). This involves the presentation of a model of some process or system to the student in the hope that by studying its performance the student will gain insights into whatever process or system is being modelled (O'Shea and Self, 1988). Often simulation may serve to remove complications which could obscure the more important principles to be understood.

In simulation training the student's role is usually more than that of a mere spectator. While simulation can and is being used for demonstration purposes, it can also cater for interactive training. Often the student is responsible for providing inputs to the model, after deciding on some strategy of use, and thereby is able to experiment with the modelled system.

Simulation is of direct importance to the training of process operators since it is the only method that enables the training of operators for characteristics of process dynamics, time dependencies, fault diagnosis, fault management etc.

An important consideration is the degree of simulation fidelity necessary to achieve the training objectives. This is thus greatly dependent upon the task(s) to be learned and the stage of learning, which in turn will specify the constraints that will determine the medium for the simulation.

For example Marshall et al. (1981) describe a low fidelity training simulation which has been "extensively developed and applied in industry to deal with fault diagnosis". Early on, faults are represented by line drawings of the display while at a later training stage a technique of "back-projection of slides of control-panel mock-ups showing different fault-conditions is used to provide a life-size representation of a panel".

Different approaches and techniques have been developed and implemented that aimed to prove the appropriate environment for simulation training. Earlier general purpose examples (Shepherd, 1986, 1989) include: (i) the manually operated Carmody Universal Trainer, which required a trainer to move instruments individually from his instructor's console, (ii) 'wet simulators' which are dedicated constructions of pipes and valves, used to boil up water and pump it around a system, (iii) electronics analogue devices, such as the Simtran II, which comprised different modules that could be fitted together in different configurations.

The technological advances in computing have resulted in the development and implementation of computer based training simulators, (examined in greater detail in section 3.6.6.) as both part-task and full scope simulators which are custom built and can offer a high degree of fidelity.

3.6. Computer Based Training

Computer based training (CBT) is a generic term that covers both computer assisted instruction (CAI) and computer managed instruction (CMI). Computer assisted instruction (also known as computer assisted training, CAT; and computer assisted learning, CAL) refers to the use of the computer as an interactive training medium through tutorials, simulations and drill and practice. Computer managed instruction (also known as computer managed learning, CML; and computer managed training, CMT) refers to the use of the computer to direct the student through a course which may, or may not, be computer based and also carry out an evaluation of the effectiveness of the training based on the results of tests and measures of performance taken during the course.

Computer based training has received considerable attention in recent years due to decreasing computing costs, the increasing flexibility of computer systems, and the increasing sophistication and flexibility of development platforms. CBT has many benefits over traditional methods that make it a very valuable tool. This is evident from the great range of training programs that have been developed both by type and by field of application. This range extents from small computer based tutorials which aim to teach basic principles to children, to full scope training simulators of process and nuclear plants where human operators can receive hands-on experience in a realistic environment.

The results of a survey by Dartt (1985) reveal the importance given by plant management to effective operator training schemes. An overwhelming majority of the managers pointed out that improved hands-on and practical operator training is much more important than the traditional methods of training in a plant environment i.e. lectures, presentations, and on-the-job training with an experienced operator. An important aspect of process operator training is considered to be the development of an appreciation of the way process systems respond to their changes. This can be achieved with the use of training simulators. This view is further supported by an earlier survey conducted by *Chemical Week* (1983) where the use of simulators as training tools was found to be very appealing and desirable.

3.6.1. Costs of Computer Based Training

The routine use of CBT, while predicted, has been inhibited by the reluctance of instructors to use it as a training method. This is mainly attributed to the associated costs that are involved both in developing the CBT material and in integrating this material with current instructional practice. In this section we pursue an analytical examination of the costs involved in the development and implementation of computer based training. These costs can be described in two categories (Dean and Whitlock, 1988), *Start-up costs* and *on-going costs.*

3.6.1.1. Start-up costs

1. People

Before CBT can be implemented successfully it is necessary for the instructors to acquire a great range of specialized skills. Assuming that all the necessary steps as described in sections 3.1. to 3.4. have been performed the training design team will now be faced with the formulation of the instructional material for CBT. This will involve familiarisation with skills of programmed learning and authorship, screen design, user-interface design and will also involve familiarisation with the authoring environment (software and hardware) that will serve as the platform for the development and implementation of the CBT course. Unless the training objectives are fairly simple and the envisaged course is also straight forward then there may be a need for a team of specialized developers. As an example we can consider two extreme cases.

The first case is that of an instructor who wants to develop a simple history simulation, such as DRAKE (Payne et al., 1980), to augment the educational value of his overall course. DRAKE is a decision-making simulation in which students are encouraged to role-play with a view to gaining an insight into a historical situation and also establish empathy with the central character. The subject of the simulation is the story of Drake's voyage around the world. It has been developed and implemented by the teacher in BASIC (programming language) and while not extremely sophisticated it has proved sufficient.

The second case is that of the development of a high fidelity full scope simulator for the training of an Olefins' plant process operators (Jones and Brook, 1990). The complexity of the problem demanded the attention of two development teams. The first team was headed by a Project Manager and comprised of Lead Engineers who were responsible for the technical aspects of the process model, the emulation of the Distributed Control System and the hardware and software aspects of the training simulator.

The second team also headed by a Project manager comprised of training, operations and process engineering specialists. In order to optimize the development effort it was necessary to set up communications protocols between the Project Managers and in addition a technical level of communication was established where technical information was exchanged.

Clearly the two examples correspond to two extreme cases of simulation training. However, both are justifiable. In the DRAKE simulation it was possible for one person to undertake all the steps necessary for formulating the subject matter as CBT material. It did not involve any extensive specialized knowledge. The teacher employed his familiarity of BASIC and his first hand experience of teaching the subject to implement a linear programmed instruction structure. Furthermore, the interactive questions and answer session was again drawn from his own experience.

In the Olefins plant training simulator this was not possible. The complexity of the problem was such that a whole range of knowledge and skills was needed. This need for specialized skills forced the management of the Olefins plant to contract an external service organization (the first team described above). Overall, the people cost involved with such an effort is great and it is not easily justifiable.

2. Equipment

It is highly likely that the introduction of CBT will lead to additional equipment costs, the least of which may be terminals or microcomputers in the training department for course development. A higher cost factor may arise from the hardware platform from where the CBT course is to be run. The two cases above are also indicative of the range in equipment costs.

The development of DRAKE was done on a personal computer while the development of the Olefins simulator required a network of computers to be used by each of the team members. Also the delivery platform was different. DRAKE was implemented on a basic personal computer similar to that which was used for its development, while the Olefins simulator required a DEC MicroVAX 3500 (TM) which was to execute all the simulation software within a DEC VMS operating system environment which was rated at 2.7 MIPS and which was provided with 16 Mbyte of memory. Further costs included specialized Trainee Operator and Instructor Consoles.

3.6.1.2. On-going costs

1. Course production

Many statistics have been produced for the time it takes, in man hours, to produce one hour of a CBT course. Estimates (Eberts and Brocks, 1987) range from over 2000 to just a few hours per hour of instruction. However, the figures (Fielden, 1979; McFarlane, 1983; Dean and Whitlock, 1988) generally vary around 200:1. This time is dependent on several factors including the characteristics of the lesson to be produced (for example: training objectives, course sophistication and duration, brand new versus existing course that needs updating etc.), and the level of experience of the instructor.

Furthermore, the development of specialized tools and software necessary for the formulation of the course may greatly increase the development effort and thus the course production costs. For example, when training simulators for the process industries are considered, major cost factors can be identified in the initial development of models for plant and control systems, of robust numerical solution techniques for simulation in real time, and of operational and training sequences.

In addition to the people costs attributed to course production (man hours to one hour of instruction ratio), there are recurrent costs of the equipment used for course development, depreciation, maintenance, paper, etc. and the costs of non-computer based material which may be used in conjunction with the CBT material.

2. Course presentation

The costs for course presentation among others include the running costs of the equipment that is used, supplementary material that is handed out to the students, the time the trainer and others have to spend running the course and helping in tutorial sessions, and creation of a suitable learning environment. The range once again varies greatly.

For example in the DRAKE simulation, the associated course presentation costs where very low. The running cost of the equipment was 'minimal'. The cost for the accompanying student booklet which was handed out by the instructor was insignificant. Finally, since the DRAKE was used as a add-on to a structured lesson it did not require special trainer time allocation or the creation of a specially suited learning environment.

At the other end of the spectrum are full scope high fidelity training simulators used for the training of process operators. Here the costs involved in course presentation can be and usually are very high. The running costs of the equipment is significant, there is a need for a full support team of trainers, need for extensive supplementary material and also a need (Ancarani and Zanobetti, 1983) for the provision of an exact replica of the control room which involves both the emulation of the control system (man-machine interface) and the replication of the work environment.

3. Course updating

Courses may need to be revised because of errors, changes in the subject matter or to improve the content. The associated costs are directly related to the approach followed in the initial development of the training (instructional) material. Other than the costs that may fall in any of the previous categories (people, equipment, course production, and presentation), there are costs associated with the direct modification of the training course. Often computer based training material become deeply embedded in bespoke hardware and software implementations making subsequent maintenance and changes difficult to implement leading to a rapid obsolescence of the entire system.

3.6.2. Benefits of Computer Based Training

The benefits achieved from the use of CBT are often "soft" and difficult to quantify in the usual accounting sense. However, in areas where CBT is appropriate, the benefits are none the less real and often quite significant (*Chemical Week,* 1983; Eberts and Brocks, 1987; Billing et al., 1986; Gran et al., 1988; Dean and Whitlock, 1988; Leitch and Horne, 1988; Elston and Potter, 1989). It is possible to describe these benefits in two categories: (a) student benefits, and (b) training department benefits.

3.6.2.1. Student Benefits

1. Consistent presentation of the training material

Computer based training courses can be formulated so that the instructional strategies of the best instructors can be incorporated and replicated in a CAI lesson. This leads to consistent high quality instruction on a large scale which is capable of continuous refinement to incorporate new elements on the basis of research or experience. (Eberts and Brocks, 1987; Leitch and Horne, 1988)

This consistent instruction and presentation of the instructional material contrasts with the more conventional training methods which depend greatly on the teaching abilities, knowledge, and availability of expert instructors. The development of a CBT course is normally a team effort, with the team being comprised both by expert trainers and by subject experts. Thus, an optimal integration of the instructional material with the most appropriate instructional strategies, needed to meet the training objectives, could be achieved.

Finally, this consistency of instruction and instructional material leads to the cost-effective "distribution of expertise to student groups, with a greater consistency of the eventual proficiency across a student group than can be achieved by conventional, time-constrained means" (Leitch and Horne, 1988).

2. Special short-term requirements can be met

In cases where either modifications to existing procedures, task descriptions, and operational objectives have taken place; or the introduction of totally new factors (such as the introduction of new equipment) affect a substantial proportion of the company's employees there is a need for them to undergo a short amount of training fairly rapidly. In such cases a short CBT module may be desirable, at least for those who need to achieve a measurable minimum level of knowledge about the changes which will enable them to stay in control until proper and thorough training is provided.

3. Practical training can be made more effective

One of the biggest advantages of CBT is that it can allow extensive hands-on training in realistic environments without the complications, risks and costs arising from either workshop-based or on-the-job training (see sections 3.5.4. and 3.5.5.). Furthermore, a lot of practical training has a very low student/teacher ratio - often one to one. While the effectiveness of this training with conventional methods greatly depends on the people doing the training, with CBT it is possible to provide consistent high quality training and retain this low student/trainer ratio.

Practical computer based training can be most effective when CAI courseware can be embedded in a computer based operational system so that the learning environment and testing situation is highly similar to the operating environment. Embedded training provides the best of two situations: hands-on training and the instructional techniques of CBT. In the process industries this is extensively supported from the use of a range of training simulators where process operators are trained to operate a plant under both normal and abnormal conditions. Of particular importance is computer based training aimed at the development of diagnostic skills through the presentation of simulated fault scenarios, alarms etc.

4. More effective use of student time

The self-paced nature of CBT and the use of screening tests that identify the levels of knowledge and expertise of the student both prior to the course and after its completion means that instruction can be provided effectively. Students with higher levels can complete the course faster than those with less or those who are slow learners, but eventually each should complete the course satisfactorily. This gives the opportunity for the better students to become productive sooner, yet the less able will not return to the job with gaps in their knowledge.

It has also been shown (Dean and Whitlock, 1988) in many cases, that the time taken for meeting the training objectives by the majority of students completing the CBT course, is significantly less than with conventional courses. "This is called *learning compression* and it is becoming widely accepted as an important factor in justifying CBT" (Dean and Whitlock, 1988).

5. Training to a qualified level of achievement and proficiency

The main goal of all forms of training is to meet the training objectives. However, it is not always possible and a number of reasons could be blamed for this. Amongst these are time constraints imposed both from the time allocated for the actual training course and also from the time afforded by the students' prior to taking up their roles at their actual place of work. On the other hand "if a formal course is necessary, the content may fill only part of a day and the temptation is to stretch it to a full day to justify bringing people from far and wide" (Dean and Whitlock, 1988). Computer based training allows for a qualified level of achievement and proficiency with "performance and attainment the end-point rather than the expiry time available for training" (Leitch and Horne, 1988).

6. Privacy

Computer based training allows students to succeed or fail in private. Thus, the embarrassment of failure is lessened, students feel free to take more risks and explore different possibilities (Eberts and Brocks, 1987). Students are often inhibited from asking questions either in class, in a tutorial or directly to a tutor. CBT allows them to ask questions and 'teach themselves' without embarrassment or intimidation (Billing et al., 1986). Furthermore, they can repeat and review sections of a CBT course until they feel confident with the instructional material.

7. Immediate feedback

Positive reinforcement and feedback form an important part of training. Furthermore, "it is most effective when it is directly and unambiguously linked to a particular action so that the student can quickly see a mistake" (Eberts and Brocks, 1987). Immediate feedback is another advantage gained from the use of CBT where the training process is highly interactive with "the student assuming the active role which has been proved optimal for learning" (Billing and Yue, 1987).

In a CBT course student questions, answers, and decisions are immediately evaluated and the appropriate feedback is provided through the implementation of a tutorial dialogue utilizing the most appropriate instructional strategy (Lysaught and Williams, 1963; O'Shea and Self, 1988; Halff, 1988).

8. Idiosyncratic adaptation.

A good CBT system will not only proceed at the student's pace, but it could also provide material in a manner compatible with the student's learning characteristics (e.g. abstract versus concrete, verbal versus graphic, concept versus example) (Eberts and Brocks, 1987). It is possible for a CBT lesson to adapt to the individual student's need thus enhancing the overall effectiveness of the training process (Sleeman and Brown, 1982; O'Shea and Self, 1988; Burns and Capps, 1988; VanLehn, 1988).

9. High student motivation

With CBT the training process is dynamic since it depends upon the student's attention and interaction. This interactive student participation and self-paced learning leads to high student motivation. It is motivating for the students as it gives the opportunity for the better ones to become productive sooner, yet the less able do not return to the job with gaps in their knowledge.

As a result of this motivation students not only exploit the capabilities of the system in a range of different ways but also use CBT systems during shifts or even for personal development (self-study) in private time, i.e. the "open learning" concept.

10. Training is available when the trainee is ready for it.

"Classroom-based training is seldom economic, and not usually very effective for very small groups (two or three people)" (Dean and Whitlock, 1988). However, it is often the case where one or two employees are ready for the next stage of their training, and have to wait until more are ready before the course can run.

Furthermore, training may be dependent upon the availability of expert instructors (Billing et al., 1986). A CBT course can be taken by the students as and when they become ready, without any delay. Thus, the motivation is retained and there is a continuity in the training process. Again trainees become productive more rapidly.

11. Reduced travel and time expenses

A flexible CBT system could be used at various on-site locations either as it is or with minor modifications. Instructional materials can be sent through communications links, thus reducing the need for costly personnel travel. In cases where offices are dispersed over a wider area, travel and accommodation costs may be a major factor in justifying CBT.

12. The student is not away from his place of work

Computer based training can provide high quality training at remote sites. Furthermore the students' testing can be ensured to be highly similar no matter what the student's location. This could lead to a reduction of costs arising from having to accommodate for students being sent to 'training centres' and, who are therefore, not available for work over a period of time. For example in the process industries operators can be trained at their plant and be readily available when emergencies or demanding situations arise (Billing et al., 1986)

13. CBT can be available at any time of the day or night

Once the CBT material has been developed and implemented on a computer system it is possible to make the material available around the clock. This could be of particular use for workers on the night shifts. Night shifts, in some industries and jobs, do not have the same amount of work as day shifts, therefore, there is a great deal of time which is unutilized. Furthermore, the computer equipment is likely to be less heavily used overnight, thus providing resources for the CBT course.

14. Training can be carried out from home

In special situations CBT could be used for self-study at home. This home based training can work with either a portable microcomputer, a terminal that can be connected into the central computer providing the training, using the public telephone system or by viewdata using a domestic television (Dean and Whitlock, 1988).

3.6.2.2. Training Department Benefits

1. Reduction in instructor involvement in a specific course

A CBT course should reduce the overall instructor involvement in the course. The CBT parts of the course should minimize if not eliminate instructor preparation time, as well as the time actually spent with the class.

CBT may also allow for a reduction in the number of instructional, administrative, and training support staff, therefore leading to significant cost reductions (Eberts and Brocks, 1987). CBT could still be cost effective in cases where courses may require a local supervisor or instructor to support or enhance the CBT material by monitoring exercises, giving feedback, discussing problems and so forth (Dean and Whitlock, 1988).

Furthermore, just as student travel costs may be reduced by CBT, so may that of the trainers if they are normally required to run courses in different parts of the country or world.

2. Amendments can be speedily incorporated

It is generally more difficult and expensive to change printed material than to make minor changes to the CBT courseware. The parts of a CBT course actually on the computer can be updated and in use again very rapidly, particularly with a centrally provided mainframe facility (Eberts and Brocks, 1987; Dean and Whitlock, 1988). A special case is that of full scope high fidelity training simulators where maintenance and modifications to the existing material may be very expensive. It is pointed out however, that it is a necessary burden since an alternative more suitable training medium which would perform their function is not currently available.

3. Easier and more accurate monitoring of student performance

The ability of CBT to monitor and record statistics on the performance of individual students including if necessary, the time spent on particular topics, can lead to considerable savings in the time of trainers in supervising, marking and recording the results of students' tests. This inherent monitoring capability of student performance can lead to the dynamic adaptation of the course structure to meet student needs, performance assessment by student and instructor and vital operational data to aid the evaluation and refinement of the system's training effectiveness (Dean and Whitlock, 1988; Leitch and Horne, 1988).

Finally, the collection of student responses and their respective paths through the CBT course, together with their recorded comments on modules and the students' performance statistics, give the trainers the ability to re-evaluate and improve the course. This function could be centralized and thus, the outcomes could be used to improve the effectiveness of the course at the same time at all the user locations.

3.6.3. Computer Managed Instruction

A Computer Management Instruction (CMI) system is one which is used for the management and administration of the instructional process. "The aim of CMI systems was to help teachers and trainers with the recording of behavioural process data and analysing it to provide a diagnosis of the state of the learning process" (Quintanilla, 1989). CMI uses the computer to direct the student through a course which may, or may not, be computer based and also appropriate the use of other instructional material or systems (for example tutorial lessons, field exercises, individual studies, instructional groups, meetings with the training staff) (Eberts and Brocks, 1987, Dean and Whitlock, 1988, Quintanilla, 1989). In effect it performs three functions (Padden et al., 1983): assessment, decision-making, and record-keeping.

Assessment of student progress relative to specific instructional objectives on a pre-learning and post-learning basis ensures that students are: a) directed towards courses that cover knowledge or skills not previously mastered, and 2) that students are not allowed to complete a course until performance criteria associated with its instructional objectives have been achieved.

Decision-making refers to the capability of a CMI system to compare student performance with 'threshold criteria' and to generate learning activity 'prescriptions' (such as (Eberts and Brocks, 1987): individual practice, text assignments, group activity, teacher consultation, laboratory sessions, a CAI session and alternative media sessions) on the basis of a student's prior accomplishments and learning style. The suggested path that the student will follow is dependent on the results of tests and measures of performance taken during the course.

Record-keeping refers to the capacity of the CMI system to collect, record, document and analyse all performance data both for students and for the instructional system generally. The records can be used for a variety of purposes (Padden et al., 1983; Eberts and Brocks, 1987; Dean and Whitlock, 1988), including:

1. The analysis of achievement of individuals and/or groups of learners
2. The analysis of test items for reliability so that it is possible to tell exactly what aspects of a course need to be improved, changed or deleted
3. The evaluation of specific learning resources relative to their comparative cost-effectiveness
4. The measurement of return on instructional investment.

It is generally acknowledged that effective and comprehensive CMI systems greatly improve the efficiency and cost-effectiveness of the instructional organization. Furthermore, it is accepted that CMI is a great improvement on alternative instructional management systems, as the demands of an effective management system are extreme. "Even relatively primitive manual instructional management systems can consume abundant human resources" (Padden et al. 1983).

However, the drawbacks of CMI arise from its very benefits. CMI is basically a management information system that requires a significant amount of data storage and power. Presently the view is (Eberts and Brocks, 1987) "that a large mainframe is a must". However, as microprocessor technology expands, micros too may have the capabilities to perform efficiently the described functions.

3.6.4. Computer Assisted Instruction

Computer Assisted Instruction (CAI) is the form of CBT which employs the computer's ability to recognize and categorize data in order to teach students by means of tutorial dialogues. This dialogue is generally controlled by the computer -which presents instructional material, makes statements and asks questions- although the student is occasionally offered options (Tawney, 1979b).

It is the norm for the computer to present theoretical subject matter - in text form, graphics displays, animation, video etc. - and then to check the student's understanding of the subject matter. This evaluation is done by comparing the students response - for example a number, a multiple choice identifier, completing a statement etc. - to conditions that have been predefined in the CBT structure. Following this internal evaluation of the student's responses the computer then provides the appropriate feedback (for example, errors may be pointed out with possible explanations, while correct answers may be rewarded) and continues with the next appropriate step in the instructional sequence.

Furthermore, CAI is often used in conjunction with other instructional systems. For simulators, CAI can be used to guide the instruction. CAI can also be used as a component of a CMI system. Finally, intelligent CAI (ICAI) is a class of instructional programs that is made more intelligent through the use of artificial intelligence (AI) techniques. ICAI can be used to control all aspects of the pedagogical interactions with little or no assistance from an instructor (Eberts and Brocks, 1987). This extension of CAI will be discussed in section 3.6.5.

In CAI the interaction of the student and the computer is highly and carefully structured since the tutorial dialogue is concerned with the organization of knowledge. As a result the student's training is performed in a constrained manner. Therefore, the designer needs to think in detail about what information is involved in a particular topic area and the order in which it should be covered so that the student's formation of links between items is optimized (Peterson, 1979).

The main criticism of CAI arises from its tutorial nature. A question often asked is: 'Would it not be better if the tutorial was carried out by a human tutor and not by a computer?' Assuming that it would be financially feasible for each student to have a one-to-one session with a tutor, then certainly some aspects of the tutorial would be better if covered with a human tutor who can embellish the tutorial session by drawing from a detailed knowledge of the subject and with examples based on his experience and from common sense. Therefore, he can adapt the training and focus directly on a particular student's needs more effectively. However, there are constraints that may limit the effectiveness of this approach (see sections 3.6.1. and 3.6.2.).

In contrast to a human tutor most CAI packages have a limited range of responses. Furthermore, issues such as student's misconceptions and inherent misunderstandings are difficult to identify and represent in a computer program, as they require intelligent insight in both the students responses as well as the sources of such problems. Therefore, it is hard to tailor a CAI system to the particular needs of any one student, while the tutor can offer a range of examples accordingly to what his *experience and common sense* indicate as being the right path to follow to correct a particular student's error.

Nevertheless, the nature of CAI allows for the possibility of effective one-to-one training. The tutorial dialogue offered through CAI packages not only allows individualisation but also forces the student to become dynamically involved in the educational process. This results in students feeling a greater degree of participation. Furthermore, "computerized tutorial dialogue has a different scale; compared to what a human tutor provides, the computer provides micro-tutoring: steps are small and individual questions frequently require low-level answers" (Tawney, 1979b). A tutor would find it embarrassing and boring to conduct a dialogue at this level. On the other hand a dialogue with a machine is bound to be less personal and intimidating. Therefore, students are more likely to repeat 'difficult' sections time and time again until they achieve the necessary and/or expected level of competence.

This is also one of the reasons why CAI is often used as an additional tool in the training process even in areas where the complete substitution of the human contact is not desired or even envisaged. The amount of computer control over the complete instructional process varies with the application. CAI can be used as an ancillary part of an instructor's lesson, providing instruction on a small subset of the total material (Eberts and Brocks, 1987).

Various approaches have been followed in the formulation of interactive CAI sessions (for an extensive analysis see O'Shea and Self, 1988). However, the two main approaches are linear programs and branching programs. These are examined in some detail in sections 3.6.4.1. and 3.6.4.2. respectively.

3.6.4.1. Linear Programs

The first methodology to be implemented in CAI was linear programming. Linear programs derived from Skinners "principles of operant conditioning, the basic law of which states that if the occurrence of an operant is followed by the presentation of a reinforcing stimulus, the strength is increased" (O'Shea and Self, 1988). The work of the behaviour psychologist indicated that the student must be taught individually in small steps and be provided with immediate feedback on his response. If he is successful, he must be rewarded ('Good!'). The important event in this approach is considered to be the reinforcement following the occurrence of desired behaviour. As a result for greater effectiveness the teaching material should be organized so as to maximize the probability of correct responses.

In linear programs (Lysaught and Williams, 1963; Atkinson, 1976; Tawney, 1979a; Dean and Whitlock, 1988, O'Shea and Self, 1988; Quintanilla, 1989) instructional material (frame) is presented to the student according to a pre-arranged sequence which has been devised with the aim of taking the student one small step towards the desired behaviour and or level of expertise and knowledge. The student is then prompted by the system for a response which is directly evaluated. Immediately after, the student is informed whether he is right or wrong (more often than not without any additional or explanatory feedback). Once this step is completed the program moves on to the next frame which has been predetermined by the author of the teaching material and is independent of the correctness of the student's response.

The main contribution of linear programming is its emphasis on the importance of feedback and individualisation. However, in this approach feedback is considered to be important only after correct responses and thus it is inflexible. Further, disadvantages of this approach arise with "its inability to characterize individual students' knowledge of specific skills, and its inability to relate students' skills to curriculum as anything more than a ratio of problems correct to problems attempted. The program cannot make fine distinctions between a student's strengths and weaknesses, and cannot present instructional material specifically appropriate to that student beyond 'harder' or 'easier' lessons" (Atkinson, 1976).

As a result, the individualisation that the student receives with linear programming, is that he may work through the material at his own pace. However, "there is no way that he may receive material different from that received by any other student" (O'Shea and Self, 1988).

3.6.4.2. Branching Programs

The lack of flexibility of linear programs in adapting to individual learning states led to the idea of branching programs (Atkinson, 1976; Tawney, 1979a; Dean and Whitlock, 1988, O'Shea and Self, 1988; Quintanilla, 1989), which use incorrect student responses to identify misconceptions and accordingly branch the program sequence so as to try and overcome these errors; for example by presenting further information.

Branching programs are based on Crowder's approach (O'Shea and Self, 1988) of 'intrinsic programming'. In this method the student controls his own progress through a lesson, by the responses he makes to the questions or problems he meets. These questions are set in multiple-choice form to enable him to select the route or branch appropriate to his understanding of the material. However, "a programme with multiple-choice questions is not an intrinsic programme unless each separate answer choice in each question leads the student to material prepared for the student who has made that particular choice" (Dean and Whitlock, 1988).

Comparing branching programs with linear programs we note that the frames in branching programs tend to be larger units with the instructional material having a wider scope. This is possible since the author is not constrained by the need of ensuring that the student responds correctly. In intrinsic programming the student is usually asked to respond to a multiple-choice question where alternative answers may be more or less acceptable, rather than totally correct on incorrect. Thereafter, depending on the students answer the program reacts with a relevant comment and either repeats the frame or moves on to the next in a predetermined sequence of frames (O'Shea and Self, 1988).

In intrinsic programming feedback serves mainly to correct misunderstandings on the student's part, and this leads to a higher degree of individualisation. As the path through the instructional material is determined by student responses, most likely different students will follow different paths. Furthermore, the extent and detail of the training session will differ, with the less able students receiving a greater number of explanatory corrections. As a result we now have an *adaptive teaching program,* which determines the sequence of instructional actions according to an individual student's performance history (Atkinson, 1976).

The main disadvantages of branching programs arise from their inherent nature. First, the author of the system has to accommodate for every possible response and also place it within the context of the CBT program both in terms of instructional relevance but also in terms of the knowledge structure. It is important that the student is not lost through branching and thus be driven away from the immediate training objectives. Another consideration is the actual characteristics of the branching options. These are usually chosen "according to stereotyped responses and not in accordance with a model of the learner, reflecting individual difficulties and learning styles" (Quintanilla, 1989).

3.6.4.3. Examples of CAI Systems

A great number of CAI systems have been developed through the years covering subjects as diverse as elementary algebra and fault diagnosis for process plants. A thorough presentation of the whole range of CAI systems that have been developed is beyond the scope of this thesis. Detailed descriptions of such systems are available in various sources; for example: Payne et al., 1980; Tawney, 1979; Eberts and Brocks, 1987; O'Shea and Self, 1988; Dean and Whitlock, 1988. Instead we shall examine some representative examples of CAI systems that are aimed at process operators and process engineering training.

1. The Problem Solving Tutor

The Problem Solving Tutor (PST) is one of three (the others being a *Safe Handling of Chemicals Tutor* and a *Scaffold Users Guide*) generic, modular courses developed by the CBT User Group of The Chemical Industries' Association, aimed at "real training needs common to the Group's participating companies and others" (*Process Engineering*, 1987).

The PST attempts to teach the user a mental discipline rather than a set of rules about a physical aspect of plant operation. Its aim is to familiarize the student with a generic approach to solving problems (the 'problem-solving algorithm': identifying all causes, ranking them in order of ease of testing, and then proceeding to identify the fault by elimination). The PST starts with an introductory session which outlines the algorithm. In this stage the user has to follow a set of well defined steps which may be applied to solve 'any problem': monitor what is happening, assess the situation, identify possible causes, and carry out checks.

Following the introductory stage the student is offered a choice of solving one of two problems, neither of which is process plant related. Instead, they are designed to exhibit the generality of the mental discipline and the proposed approach. The choice of one of the problems initiates a structured presentation-question-answer session which serves as the training medium.

In addition to the computing environment each user is also given his own printed work book which accompanies the lesson. This gives background and explanatory information and a complete flow diagram of the problem-solving algorithm. In order to get good results trainees need to refer to the book so both tools are utilized effectively. When necessary, the program ensures that the student refers to the book by inquiring for a specific keystroke which will allow it to proceed 'to the next page'. This is only given in the printed text.

One of the two choices involves 'fixing a faulty kettle'. In this problem the user is initially given the scenario and is asked to list all the possible reasons which would explain why the kettle is not working. Once the student is ready he can compare his list to a list of 13 reasons provided by the PST. Having deliberated on the possible 'faults' the user is now asked to rank them with respect to plausibility. This ranked list is then examined by the system through a correlation algorithm which also suggests where the user might have gone wrong. The appropriate feedback allows the student to eventually draw up a sensibly ordered list.

The final step in the PST tutorial session involves carrying out test(s) which will help the student eliminate one or more of the possible causes. These tests are performed interactively with the computer in a 'dialogue' and the results of these tests are presented on the screen (e.g. the wall switch is on). Upon successful completion of the various tests the session ends with a brief conclusion as to why the hypothetical kettle has failed.

2. Start-up Procedure for an Ammonia Plant Compressor.

The *start-up procedure for an ammonia plant compressor* (SPAPC) program was developed and implemented by ICI Severnside Works (Billing et al., 1986) as part of a wider plant to introduce on-site facilities for Open Learning. The instructional mode employed in SPAPC is that of drill and practice. The student is presented with a series of examples aimed at helping him develop proficiency in a specific skill. Furthermore, SPAPC provides the 'necessary guiding feedback' to facilitate learning of the particular skill.

The compressor start-up procedure involves a sequence of nineteen operations. The operators need to be fluent with this sequence so that they can perform the start-up quickly if necessary. During the first training stage the program takes the student through the nineteen steps with prompts. At a second stage, the student is asked to interactively repeat the start-up procedure with as few prompts as possible, and is given feedback on performance. The aim is to complete the start-up without asking for more than five prompts.

3. Introduction to the Ammonia Plant

This has also been developed and implemented by ICI Severnside Works (Billing et al., 1986). This is an extension of traditional book-based programmed learning. The *Introduction to the Ammonia Plant* (IAP) mainly uses text and graphics to present information to the student. Occasionally, animation is also used which helps make the presentation more dynamic and effective.

The program describes the main units in the process and what happens to the flow as it passes through the plant. It also provides suitable feedback, in some cases including remedial information, and is able to adapt to the student's performance when appropriate by using branching methods. In the course of the tutorial the student is asked questions to test understanding and determine the necessary feedback, and a multiple choice quiz is given at the end to ensure that the important points have been grasped.

4. License Pursuit

Licence Pursuit was been developed by the Westinghouse Electric Corporation (*Nuclear Engineering,* 1988) as a supplementary training tool which could be used for self-study. Patterned after the game *Trivial Pursuit™, Licence Pursuit* allows up to four students to compete with each other. While they are competing they are learning aspects of thermal science and reactor physics.

When they answer a question correctly, the answer is supplemented by additional information; when they answer a question wrongly, the correct answer and additional information are given. When the game ends, it is claimed that "the students not only know more than when they began, but also know where they are weak and must focus their future studies" (*Nuclear Engineering,* 1988).

5. Induction Training

Marshall (1988) introduces an *Induction Training* program that has been developed by BP Chemicals Hull Works. The course was designed to provide the trainees with thorough information on safety alarms and procedures before a supervisor takes them physically around the plant to indicate the physical location of real items. The system utilizes an Interactive Video Scheme in order to present information realistically. The scheme is an integrated environment of video images, computer graphics, explanatory text and related questions.

It first presents information via both images and text, and then asks questions. Two types of questions are supported. The first type is regular multiple-choice questions. The second type is a more flexible type were the operator has to type in keywords. The system also provides appropriate feedback according to whether the question was answered wrongly or correctly. Branching is also supported. If erroneous answers persist the trainee is directed to sections that should be revised.

3.6.5. Intelligent Computer-Assisted Instruction

Intelligent Computer-Assisted Instruction (ICAI) refers to the use of artificial intelligence (AI) techniques for the enhancement of computer based training and instructional systems. "AI techniques using human intelligence as both a model and a source of data, instruct students by interacting with the student naturally, answering questions, and providing a data structure so that facts can be acquired and retrieved efficiently" (Eberts and Brock, 1987)

ICAI systems (also referred to as *Intelligent Tutoring Systems*, ITS) represent an evolution of the more conventional CAI systems (Section 3.6.4.) with a view to providing each individual student with a fully adaptive learning environment. The goal (Sleeman and Brown, 1982; Eberts and Brocks, 1987; O'Shea and Self, 1988; Quintanilla, 1989) of such a personal tutor (tutorial system) is to enable a student to build on his existing knowledge and/or level of expertise, utilizing the individual's learning abilities and learning style preferences and adapting to his motivational state, needs and learning rate.

In contrast to more conventional adaptive CBT systems "ICAI programming is generative; it can be run repeatedly by the same student although his learning situation is different each time. Thus, ICAI is not completely dependent on programming that attempts to account for every possible interaction future students may require" (Eberts and Brocks, 1987).

ICAI is targeted toward complex learning domains that require flexible instructional strategies and delivery methods. This is expressed through a set of particular benefits provided or envisaged by the application of such systems. First, the instruction delivered by the system can be very flexible. An intelligent system can use its own 'intelligence' to handle a situation that has not been anticipated by the course developer. Second, training is tailored to the individual student at the particular instance of his interaction; most ICAI programs try to understand the individual student by deriving a model of what the student does and does not know. Third, the student is allowed a more interactive experimental role in the learning process, with emphasis placed on providing situations in which the student can query the computer to try to discover the correct answers. "ICAI programs are designed to direct, not lead" (Eberts and Brocks, 1987).

For the above benefits to be achieved an ICAI system needs three main components which underline its functionality. Burns and Capps (1988) defined them as the "tests of intelligence". First, the subject matter, or domain must be represented in the system in such a way, as to allow the expert tutor to draw inferences or solve problems in the domain. Second, the system must be able to assess the student's level of understanding and his progress through the learning process. Third, the "instructor in the box" should be capable of implementing as required, a range of instructional strategies in view of reducing the difference between expert and student performance. These three components are, respectively: *(a) The Expert Module, (b) The Student Module, and (c) The Tutor Module.*

Finally, for the ICAI to be effective there is also a need for a fourth component which allows for student and human instructor interaction. This *Communications Module* provides the instructional environment and the human-computer interface through which the tutorial is administered, with the appropriate dialogue format.

3.6.5.1. The Expert Module

The *expert module* (also referred to as the *domain model)* contains the domain knowledge of the subject to be taught and as such provides the backbone for the development of ICAI systems. Therefore, any *expert module* has to include an abundance of specific and detailed knowledge relating to the instructional material. The sheer amount of knowledge required in complex domains ensures that developing the *expert module* may be the most demanding chore in building an ITS (Anderson, 1988; Burns and Capps, 1988). This knowledge -as with other non-ICAI *expert systems*[1] applications- is normally retrieved from people with years of relevant experience in the particular domain. Two approaches are most common for modelling

---

[1] For a description of Expert Systems techniques and building tools the reader is referred to Hayes-Roth, Waterman and Lenat (1983); and Gevarter (1987) Harmon (1987). For expert systems application and their relevance to process control the reader is referred to Moore and Kramer (1985); and Zimolong, Nof, Eberts and Salvendy (1987) respectively. Finally, for an overview of expert systems applications in Process Systems engineering the reader is referred to Stephanopoulos (1987a) and Niida, Itoh, Umeda, Kobayashi, and Ichikawa (1986).

the expert, each moving toward a more "cognitively faithful representation of the content expertise" (Burns and Capps, 1988).

The first kind of model is called a *black box model.* This model involves the representation of the domain knowledge without actually codifying the underlying human intelligence. For example, a system can use mathematical equation-solving techniques to obtain the result of the reasoning while humans will achieve the same through symbolic processes. For example, SOPHIE (intended to teach students how to troubleshoot faulty electronic circuits; Brown, Burton and Bell, 1975) used a general-purpose electronic simulator called SPICE II to check the consistency of student's hypotheses and answer some of his questions. Its mechanisms are never revealed to the student since they are not part of what the student is expected to learn. However, this dependency on the simulator made it impossible for SOPHIE to explain its decisions in detail. Therefore, the simple input-output information available from the black box system was considered inadequate for instruction.

One method of enhancing these models is to employ a methodology called *issue-based tutoring* (Burton and Brown, 1982), where issues are identified as the important aspects of the domain (that is, the skills and concepts the student is expected to master). With *issue-based tutoring,* instructions are attached to specific issues characteristic of the behaviour of the expert and the student within the learning environment. In this way, when a student chooses (or fails to choose) a behaviour, he or she may receive feedback about the particular behaviour. For example, if the expert modelled chooses to perform a certain action while the student does not, the system will interrupt the session with an explanation of the usefulness of that particular action.

The second model for representing domain expertise is the *"glass box" or articulate model.* This model is referred to as "articulate" because each problem-solving decision it makes can, in principle, be explained in terms that match (at some level of abstraction) those of a human problem solver. This model attempts to enable the ICAI system to simulate the human problem-solving process. Therefore, since the pursued solution path of the system is analogous to the human one, the system can help the student by explaining and analysing single problem-solving steps. For the system to be able "to simulate human problem-solving, it needs to have procedures for knowledge acquisition, reasoning, planning and problem solving and to generate and test hypotheses" (Quintanilla, 1989). The WUMPUS (Goldstein, 1982) and GUIDON (Clancey, 1982) systems are based on articulate experts, as are many production rule-based experts.

When representing knowledge in an ICAI system, the information must be organized so that acquisition, retrieval and reasoning can be performed efficiently. Three types of knowledge are represented in present ICAI systems: *declarative, procedural and heuristic knowledge;* with the first two being more commonly used (Eberts and Brocks, 1987; Woolf, 1988; Anderson, 1988; Burns and Capps, 1988). These differ mainly "with regard to which aspects of the database are explicitly represented and which aspects must be inferred by the organization of the data" (Eberts and Brocks, 1987). Woolf (1988) describes these three types:

Declarative (or Conceptual) knowledge includes the data, concepts, and relation between the concepts in the domain. This knowledge has traditionally been the primary domain knowledge represented in tutoring systems. In many systems concepts are represented by a frame or other data structure that encodes default values within an explicit set of attributes for each concept. Such a data structure expresses information about both the attributes of a concept and the relationship between concepts.

Procedural knowledge includes the reasoning used by the system to solve problems in the domain. This knowledge has traditionally been included only in teaching systems that reason about procedural tasks, such as solving arithmetic problems or simulating the operation without a tutor for advising the student about his interaction with the simulation.

Heuristic (or qualitative) knowledge includes actions taken by an expert to make measurements or perform transformations in the domain. This knowledge has rarely been included in tutoring systems, but must be included if tutors are to monitor their students' problem-solving activities. Heuristic knowledge defines the operations performed to solve problems in the field and is part of an expert's experiential knowledge about how to obtain answers. It differs from procedural knowledge in that it does not add content to the domain, nor does it represent *concepts in the domain;* rather, it describes *how to solve problems* and the actions taken by the expert in using the conceptual and procedural knowledge.

In general, two broad categories are used in present ICAI systems. *Declarative representations* are explicit about what is represented in the knowledge base. How the knowledge is used is less clear. *Procedural representations* are explicit about the step-by-step use of knowledge. However, it is difficult to determine what is or is not known from the knowledge that is represented. What is known must be inferred from the behaviour of the system. It is also pointed out that information organized one way can also be organized using the other method.

These procedural representation methods describe information such as the constraints of the system, the state of the system, critical events that will change the state, boundary conditions, and how the system will transition from one state to another. Example methods[2] used for knowledge representations include: production systems, semantic networks, finite state automata, procedural networks, and augmented transition networks.

---

[2] We abstain here from an extensive examination of each of the knowledge representation methods. Even though it is an important issue to the development of ICAI systems it is beyond the scope of this thesis. The reader is referred to the AI literature for further information (for example Charniac and McDermott, 1985). For AI in the framework of Process engineering the reader is referred to Stephanopoulos (1990).

3.6.5.2. The Student Module[3]

The use of a *student module* is another feature that distinguishes ICAI from CAI systems. Although, not used by all ICAI systems, the use of one enables a system simultaneously to be powerful and flexible (Eberts and Brock, 1987; Burns and Capps, 1988). The student module is used to represent and asses the user's understanding or progress in understanding the domain model. It infers from the student's behaviour what the student knows. Therefore, it can selectively present information that the student does not know and correct the student's misconceptions. Thus, "instruction can be more individualized than that available on traditional CAI systems" (Eberts and Brock, 1987). In a student module the knowledge structure that depicts the student's current state is the *student model* and the reasoning process to develop it is called *student diagnosis*. Five general methods are currently used to determine what a student knows.

The first two methods used, *topic marking* and the *context model*, are relatively simple approaches. For the topic marking approach the system keeps a record of the information with which the student has been presented and can therefore avoid unnecessary repetitions. For the context model, the extent of the students knowledge is interpreted in terms of the dialogue and questions that the student asks the system. The possible feedback provided by the system is categorized and thus, the level of the dialogue maps on predefined representations that are used to classify the students progress in the learning process.

Two other methods, the *bug and overlay approaches*, compare the student's knowledge with the knowledge of an expert. For the bug approach *(student bug*

---

[3] "Student modelling is, in general, a difficult area and there appears to be little consensus of approaches in this area" (Leitch and Horne, 1988). Therefore, we limit the presentation of this component of ICAI to a brief summary of descriptions available in the literature. In particular the sources for this were: Sleeman, 1982; Goldstein, 1982; Eberts and Brock, 1987; O'Shea and Self, 1988; Clancey, 1988; Payne, 1988; Gilmore and Self, 1988; VanLehn, 1988; and Quintanilla, 1989.

model), student knowledge is characterized in terms of the bugs and misconceptions the student has about the subject compared with an expert's knowledge. Student performance is compared with a variety of bugs until the best match is found and thus determine which incorrect model is consistent with the student's behaviour. For the overlay approach *(overlay model),* student knowledge is characterized in terms of a subset and simple variations of the expert's knowledge. Once the student demonstrates the correct facts or rules contained in the expert's knowledge, those facts are considered as having been acquired by the student.

The fifth method is the *generating modeling* approach. Instead of concentrating on particular facts that may be or have been acquired, the student's knowledge is defined in terms of the plans used to solve a particular problem. The computer instruction is then organized according to these perceived plans; factual misconceptions are corrected and feedback is tailored to the particular student's conceptions.

3.6.5.3. The Tutor Module

The tutor module controls how and in what order the domain concepts should be introduced, and monitors the student's progress in solving domain problems (Leitch and Horne, 1988; Quintanilla, 1989). The tutorial interactions of an ICAI system must exhibit three characteristics (Burns and Capps, 1988): (a) a tutor has to select and sequence the material to be presented, (b) a tutor must be able to respond to students' questions about the subject matter, and (c) a tutor must be able to determine when students need help in the course of practicing a skill and what sort of help is needed. Two main categories of tutors (Halff, 1988) can be distinguished in ICAI: *(a) expository tutors, and (b) procedure tutors.*

Expository tutors, are primarily concerned with declarative knowledge and inferential skills. They employ dialogue, as their main instructional tool, to teach students a body of factual knowledge and the skills needed to draw direct conclusions from that knowledge. *Procedure tutors,* teach skills and procedures that have application outside of the tutorial situation. "While memory for facts is important for learning such skills, tutors of this kind are much more concerned with the procedures that operate on memory" (Halff,1988). As a result, procedure tutors function much more like *coaches*. They present examples to exhibit problem-solving skills, and they pose exercises for purposes of testing and practice.

The problems (Halff, 1988) associated with the selection and sequence of instructional material indicate the differences between these two tutors. For expository tutors, the problems are those of maintaining focus and consistency and of progressing through the subject matter in an order that supports later retrieval of the concepts being taught. In addition to these, procedure tutors also have the problem of properly ordering the subskills of the target skill and selecting exercises and examples to reflect that order.

Tutors may use one or more instructional strategies. Strategies may range from simply following a sequence of actions which have been pre-specified by the author of the teaching program, to executing "a complex decision procedure" which attempts to take into account a student model and the course objectives (O'Shea and Self, 1988). On the other hand a tutor may allow for *learner control* (that is, making the learner responsible for making instructional decisions) in which case a tutorial strategy may be non-existent. Examples of this case are straightforward simulations.

Existing ICAI systems utilise three basic approaches to instruction: *the Socratic method* (Clancey, 1982; O'Shea and Self, 1988; Anderson, 1988; Leitch and Horne, 1988), *the discovery method* (Burton and Brown, 1982; O'Shea and Self, 1988), and *the coaching method* (Brown, Burton and deKleer, 1982; Genesereth, 1982, Leitch and Horne, 1988). The *Socratic method* provides a sequence of questions designed to guide the user through the learning process, whereas the *discovery method* centres on giving the student carefully chosen examples which increase the likelihood of his discovering a particular rule. Lastly, the *coaching method* allows the user to learn by experimentation (also known as 'reactive learning' or Experienced Based Learning).

It is important to decide on the appropriate instructional strategy or appropriate combination for the development of an effective instructional design. This choice will be defined by the nature of the instructional material and the training objectives. Table 3.1. is a summary[4] of the various issues that may point out the final characteristics of the tutor module and the way they relate to CAI and ICAI in general. These can be broken down into the following variables: individual differences, knowledge of results, amount of practice, augmented feedback, part-whole tasks, adaptive instruction, conceptual representations, and motivation.

3.6.5.3. The Communication Module

While the student is not concerned with the internal workings of an ICAI system he is concerned with how the tutor communicates with him. The communications module manages the overall human-computer interface to the ICAI system. It is the part of the system that specifies or supports the activities that the student does and the methods available to the student to do those activities (Burton, 1988). "It is tautological that the communication should be in a natural language, where 'natural' means that the language is well-suited for discussing whatever topic it is. The student should not be distracted from the subject at hand by having to search for ways to express himself" (O'Shea and Self, 1988). However, while this would provide for the ideal learning environment, it has not yet been achieved. Nevertheless, research in the area of 'natural language interfaces' (for example, Frost, 1987) provides scope for some optimism.

---

[4] For a more thorough examination of these issues the reader is referred to Eberts and Brocks (1987).

Table 3.1. Computer-based instruction issues

ISSUE	DESCRIPTION
Individual Differences	Individuals are inherently different. Therefore, The correct learning strategy as dictated from the person's abilities has to be chosen. It is important to evaluate individual differences and set appropriate task difficulty levels.
Knowledge of results	Knowledge of results is defined as the feedback or reinforcement that is given to students to tell them how well they performed on the task. Feedback should be immediate and should be used to target misconceptions.
Amount of Practice	This is based on the rationalethat *more training results in better operators*. As a result one of the best determinants of the amount of skill students obtain is the amount of practice they have performing the task to be learned. Experts are usually distinguished from novices in that they have had more practice performing a particular task.
Augmented Feedback	Augmented Feedback is defined as experimentally or system-provided cues that enhance the intrinsic task-related feedback. These cues emphasize an event so that it has a lasting effect.
Part-Whole Instruction	Part-whole training instruction is concerned with the issue of whether it is more beneficial to break a task down and train component parts (part training) or whether it is more beneficial to train the whole task all at once (whole training). The choice is dictated by the nature of the subject taught. Part learning is necessary when the amount of learning is large; while whole learning is better if the sequence is long and complex.
Adaptive Instruction	Adaptive instruction is defined as a method of individualized instruction in which task characteristics are automatically changed from an easy or simple form to the difficult or complex criterion version.
Conceptual Representations	Determining how people conceptually represent information is important for instruction because the material to be learned must fit in with how a person eventually comes to conceptualize the problem; the material must fit into the person's memory structures.

Finally, guide-lines in designing the Human-Computer Interface (Dean and Whitlock, 1988; Miller, 1988) along with the popular criteria in software ergonomics are relevant to the communication module of ICAI systems. In summary, "the system should be transparent, controllable by the user, reliable (have a high tolerance to errors), acceptable, self-explanatory, especially in the beginning phase (easy to learn) and should have a knowledge- and context-dependent help system" (Quintanilla, 1989).

3.6.5.4. Examples of ICAI Systems

A number of ICAI systems have been developed. However, as Leitch and Horne (1988) indicate, in contrast with more conventional CAI systems -which were developed largely by educational researchers and industrial training developers to solve practical problems- ICAI systems have been developed mainly by computer scientists to explore the use of AI techniques in the process of learning and teaching in the education domain. Therefore, the focus of development has been on technical aspects such as knowledge representation and natural language interfaces, rather than on the exploration of instructional strategies or domain characteristics.

The majority of such systems have not been developed past the point of academic interest and as a result they have not 'left the lab' to be integrated with a 'real instructional process' (Johnson, 1988). Furthermore, the domains that were exploited, while complex, were ones of a restricted scope that could 'easily' be represented.

Consequently, tutors aimed at industrial environments and in particular the process operator have been scarce and at a lesser level of sophistication than that achieved in ICAI systems concerned with other subject areas. The reason for this is that the process industries pose large and complex problems that are not easily represented. Consequently since the area of ICAI is still in its infancy other subject areas are more accessible to research and development.

Nevertheless, in this section we shall focus our attention to ICAI systems that have been developed with the view to be used by the process industries as training tools.

1. STEAMER

STEAMER (Hollan, 1984; Hollan, Hutchins, and Weitzman, 1984) is used by the U.S. Navy to train engineers and operators about naval steam propulsion plants. STEAMER provides graphical display and control of a simulation of a steam plant (Burton, 1988). This simulation of a steam propulsion plant consists of a graphical interface to a mathematical model of the plant.

Instruction in STEAMER (Walkers and Stevens, 1986) is geared towards teaching the student conceptual knowledge so that the student can mentally simulate the operation of the system. The intelligent tutor is used to help the student conceptualize the operation of the system. The tutor answers questions, provides hints, and gives explanations. It analyzes the student's misconceptions and guides the student toward learning the underlying principles of propulsion engineering operations (Eberts and Brock, 1987).

STEAMER allows the student to view the operation of a steam plant from external, internal, or mechanistic points of view (Burton, 1988). The interface allows the user to select from a library of views of the propulsion system and to interact with a selected view to change the state of the underlying simulation model. Furthermore, the use of graphic displays in STEAMER emphasizes different structural relations among the steam plant components.

Several levels of detail of the propulsion plant can be depicted in different views. The level of detail can vary from gauges and dials to schematic diagrams. For example, one of the approximately 100 displays shows (Burton, 1988), using animation, the flow of fluid through the plant. Another display shows dials that give the pressure at various points in the plant. A third display shows a graph of pressure as it changes through time or indicates the rate of change.

The different displays are used to demonstrate different properties of the plant's operation. The flow animation may be used to indicate cause and effect relations between different parts of the plant, whereas processes that depend on the rate of change in pressure are best demonstrated by the graph. The displays allow the student to "view abstractions of the processes involved in steam plant operation in a manner not possible in a real steam plant, and to observe in one place information that may be spread out over an entire ship" (Burton, 1988).

Many of the different types of gauges and valves that are used in the steam plant representation can be directly manipulated by the student (Miller, 1988); for example valves can be closed by clicking the system's mouse on them rather than issuing an explicit command like CLOSE VALVE-17X. Similarly, when gauges display values associated with parts of the system under the student's control, such as the temperature of a boiler, the student can reset the value by clicking the mouse on the needle of the gauge, and moving the needle to the desired value. The corresponding variable in the simulation model is then given that value and the model is updated to reflect the modification.

The main characteristic of STEAMER is that it utilizes a simulation of the spatial representation of the system (Eberts, 1985). "A central frame representation contains a large body of essentially static information which includes a formal description of the objects, principles and procedures that go to make up a steam plant in a particular state. Along with this store of knowledge, there are a variety of specialized program modules which support the use of the knowledge in a variety of applications... The second kind of representation in the system is that of the actual system being modeled and the specific procedure being executed... Around the central representation of concepts is organized a set of computational and inferential modules which embody the system's ability to act in various application environments. These include modules for representing graphic and textual views or descriptions, procedure monitoring and configuration checking components, and simulations of the operation of the steam plant..." (Walker and Stevens, 1986).

A criticism of STEAMER arises from the fact that while it knows a great deal about the mathematical properties of steam, it knows rather little about how to operate a steam plant (Anderson, 1988). As a consequence, STEAMER provides only part of the instruction necessary to operate such plants. Nonetheless, it is judged to provide an important component of the instruction.

2. *RECOVERY BOILER TUTOR*

The *Recovery Boiler Tutor* (RBT; Woolf, 1988), is a tutor built for a kraft recovery boiler, which is a type of boiler found in paper mills throughout the United States. RBT provides multiple explanations and tutoring facilities tempered to the individual user, a control room operator. The tutor is based on a mathematically accurate formulation of the boiler and provides an interactive simulation complete with help, hints, explanation and tutoring.

The RBT tutor was designed to be a partner and co-solver of problems with the operator, who is encouraged to recognize the effect of his actions and to experiment with multiple explanations of an emergency. The main hypothesis of the developers was that trainees in an industrial environment must learn to evaluate their own performance based on its effect on the industrial process. In this sense the program allows for extensive 'learner control' and also learning by discovery.

A student can initiate any of 20 training situations, emergencies, or operating conditions, or he can ask that an emergency be chosen for him. He can also accidentally trigger an emergency as a result of his actions on the boiler. Once an emergency has been initiated, the student is encouraged to adjust meters and perform actions on the simulated boiler to solve the emergency. By comparing operator actions with ones specified in the tutors knowledge base, the system can recognise optimal, less than optimal, and clearly irrelevant actions. This enables the tutor to offer help, hints, explanations, and tutoring advice when needed or requested. As a principle the operator is expected to observe the impact of his actions on the simulated boiler and to react before the tutor advises him about potential problems.

The operator's interaction with the tutor is via a dialogue format executed through a hierarchy of menus. For example, one menu allows an operator to select a physical activity to be performed on the boiler, such as checking for a tube leak or rodding the smelt spout. A second menu allows the operator to select a particular computer screen, such as the alarm board or control panel board.

The operator is given a choice of a range of possible views of the boiler with the ability to focus on several components. Assistance is also provided in the form of visual clues (a darkened smelt bed), acoustic clues (ringing alarm buzzers), textual help, explanations and dialogues. Furthermore, the operator can examine the state of individual process parameters, change setpoints, and receive various meter readings, physical and chemical reports, and dynamic trends of variables.

In addition to providing information about the explicit variables in the boiler, RBT provides reasoning tools designed to aid a student in reasoning about implicit processes in the boiler. One such tool is composite meters which record the state of the boiler using synthetic measures for safety, emissions, efficiency, and reliability of the boiler. These meter readings allow the student to make inferences about the effect of his actions on the boiler using characteristics of the boiler.

Finally, each student action is recorded in an accumulated response value, which reflects an operator's overall score and how successful or unsuccessful his actions have been and whether actions were performed in sequence with other relevant or irrelevant actions.

3. COMPUTER OPERATOR ADVISING AND TRAINING SYSTEM

Leach (1988) described an ICAI system, the *Computer Operator Advising and Training System* (COATS) developed by Texas Instruments for training process operators. The actual domain in COATS is represented as a decision tree of pre-programmed heuristics. COATS can adapt to the requirements of the trainee and either provide an introductory course or a refresher course.

A pre-test given in the beginning of the course is used to identify the operator's level of expertise. However, at this level the questions it poses relate to the actual physical experience of the operator rather than on the actual knowledge and level of expertise that the operator has. Therefore, it is based on the understanding that the operator will provide honest answers to the question.

Furthermore, it was pointed out that the system has the capability to "more or less understand" the level of expertise of the operator -perhaps through the implementation of a student model- while delivering the instructional material. It is therefore in a position to retract any initial false decisions that could have occurred if for example there was a mismatch between the level of knowledge attributed to an operator with a certain amount of physical experience and his actual level of knowledge and skills.

The training is delivered through a dialogue via a question-answer scheme. However, this communication does not allow for natural language communication. Instead, it follows a multiple choice format under which the operator is provided with a set of possible answers to choose from. However, there is a provision in the system for providing feedback and explanations. This allows the student to inquire why his answer was correct or false and possibly view additional information.

3.6.6. Training Simulators

Training simulators represent the category of computer based training most widely applied in the process industries. In a simulation the computer is acting out an application of a theory in order that the student's understanding of the theory is deepened. In this respect computer simulations can often be considered to be a replacement of laboratory exercises (Tawney, 1979b).

Assuming that the process can be modeled effectively, the use of process simulators for training offers a number of instructional facilities unique to the medium. Most notably, simulators enable 'hands-on' realistic operation of the process giving the operator a *better feel* for the whole systems. Second, process simulators provide a risk free environment to train in normal, abnormal situations, fault management and accident handling. Third, a simulator enables the trainer to expose the trainee to cases that would be impossible in the real-world but may be considered useful in terms of training.

In addition to that, the various exercises can be repeated time and time again until the student has mastered the concepts and/or achieved the required level of competence. To this extent computer simulations provide individualized training. Last, an effective computer based training program that is based on a simulator could lead the operators in following the same procedures for start-up, shut-down, and similar approaches in maintaining the process status.

The justification for simulations used in training is based upon the notion of *learning by doing*, forming concepts through experience. Smith (1979) proposes that: "Computer simulations should principally play the role of reinforcing the student's grasp of fundamentals, much in the same way as traditional laboratory experiments serve this function." The benefits of using training simulators should thus be evident.

3.6.6.1. Functional Definition[5]

Flexman and Stark (1987) in an extensive presentation of the whole issue of training simulators attribute to them two primary functions. "First, to present information like that associated with some real system for which training is required. The simulator stores processes, and displays information reflecting the functional characteristics of the system, the effects of relevant environmental events, and the effects of control inputs made by the operator. Second, training simulators incorporate special features that facilitate and enhance their ability to support practice and learning for the express purpose of influencing operator performance in the real system which is simulated."

Training simulators have six primary characteristics that differentiate them from other training devices and equipment:

1. Synthetic

Training simulators are synthetic in the sense that they are constructed only to provide task information rather than supporting real operational action functions.

2. Data Storage and Processing

Simulators store data that represent the dynamics of the system being simulated and the task-relevant portions of the environment in which the system (and the system operator) must perform. When a control input is made, its effects are displayed only after the computer identifies the parameters effected and computes the effects of that control input in relation to other control and environmental conditions.

3. System Dynamics

The response of a system to a control input or to some external influence is a function of many factors whose precise influence is important to the student learning to understand, control and employ the system in its assignment.

---

[5] The development of this and the following section on the 'Functions of Training Simulators' (3.6.6.3.) are based on the presentation of the issues to be found in Flexman and Stark (1987).

4. Controls and Displays

The simulator incorporates controls and displays that are as real as is possible, to support learning at both the intellectual and psycho-motor levels. All elements of the interface with the operator are represented, to the extent that they are expected to influence the learning of assured perceptions and their use in system operation.

5. Whole Task Support

Simulators are designed generally to support whole-task rather than part-task training. This permits the student to practice in the workload, stresses, and time pressures typical for the job for which they are are being trained.

6. Instructional Control

In essence, the training simulator contains a mathematical model of the system for which training is required and also incorporates an instructional model that provides the control necessary for learning to take place in the most efficient manner. The instructional model could be implemented either through direct intervention of the instructor or by a computer program.

3.6.6.2. Architecture of Training Simulators

The architecture for training simulators most widely employed involves primarily the use of two components: a dynamic simulator and a control system (or a control system emulator). The dynamic simulator calculates the transient behaviour of a plant over time, by solving the systems of differential and algebraic equations that describe the simulated system/process. The control system component acts as the user interface to the simulated process and permits the manipulation of process inputs and possibly the configuration of control strategies.

From a general point of view, dynamic simulators employ four basic components. These have been previously described (Helget et al., 1990) as: the user interface, the software, the algorithms and the models. The given definitions are as follows:

Models

These provide the basis for analysis and consist of physical relations usually derived from conservation, thermodynamic and rate laws. The models contain the actual chemical engineering knowledge. They must be appropriate, rigorous, and accurate.

Algorithms

These solve the model equations to produce the required results. Computational efficiency and robustness are the key requirements for algorithms.

Software

Software includes everything necessary to build and use the actual simulator. System architecture, data structures, the file system, and the programming language are considered part of the software. The portability between different hardware platforms is very important.

User Interface

Through the *user-interface*, engineers communicate with the simulator. From daily practice, they are accustomed to thinking in terms of process flowsheets, so the user interface should reflect their practice and their process-equipment-oriented approach. Ideally, engineers should not directly encounter the software and the algorithms; they should work directly with the engineering models and process flowsheet.

The inhibited use of training simulators can be explained by the difficulties which arise in the development and implementation of the above components. The arising cost factors have been discussed in previous sections of the thesis. As a result the production of such systems has traditionally "been a specialist business, and special model building tools have been developed and used by its practitioners" (Perkins and Barton, 1987).

Nevertheless, increased interest in dynamics has led to considerable research and currently the first generation of dynamic simulation tools are available some of which commercially. The main motivation for model building and simulation tools is the reduction of the development and implementation costs. Table 3.2. lists some of these dynamic simulation packages along with references for more information.

Table 3.2. General Purpose Dynamic Simulators

NAMES	REFERENCES
DIVA	Kroner et al. (1990)
GEPURS	Shinohara (1987)
QUASILIN	Hutchison et al. (1986)
SPEEDUP	Pantelides (1986)

However, these tools are not yet in widespread use, despite the many benefits that they offer and their many possible applications (Perris, 1990). This limited use appears to result from:

1. Lack of engineer-oriented general purpose models and user interface.
2. Difficulties in dynamic modeling; lack of modeling tools, skills and experience.
3. Difficulties in initializing and solving the model.
4. Unacceptably high computational effort required for industry scale problems.
5. Difficulties in handling problems which combine discrete and continuous events.

All these problems increase disproportionately with problem size, thereby limiting the usefulness of dynamic simulation in practice. As a result there is an upper limit to the size of a single manageable simulation chunk. Nevertheless, a trend towards integration of simulation software has encouraged the use of general purpose flowsheeting systems for real-time simulation.

This should have the effect of decreasing significantly the effort and costs associated with the development and implementation of dynamic simulation problems. Thus the possibility arises for defining a training simulator architecture that may be achieved in this way (Herman et al., 1985; Macchietto et al., 1987; Kassianides and Macchietto, 1990; Kroner et al., 1990;. This is further examined in the rest of the thesis.

3.6.6.3. Functions of Training Simulators

Simulators have unique training-relevant capabilities that permit them to perform additional functions which make them superior settings for training, practice and learning. The simulator performs some functions about as well as they would be performed in the actual environment. It performs some much better than they would be performed in the real-world system, and it performs some that are impossible in the real world, but which facilitate learning. Some of the unique instructional functions of simulators follow.

1. Briefing and Demonstration

Depending on the facilities available briefings can be provided in a number of different ways in the simulator. They can be provided either by an instructor following a standardized format imposed by the simulator, or by triggering one of a variety of peripheral devices that provide automatic briefing. Simulators also facilitate demonstrations. These again can either be automated or provided by the instructor. The major advantage of demonstrations is that they show the student how a task is performed. Furthermore, in the case of automated demonstration, the student can follow the task completion procedure time and time again without deviation.

2. Practice

The simulator supports practice by providing the controls and the information used by the operator in performing his assigned tasks. Its value as a practice setting is due to its ability to represent precisely the conditions needed to practice each task and each task element separately if advantageous. With this unique ability of presenting exactly the training setting needed it facilitates learning at each stage of the student's development.

3. Performance Analysis

Performance on the job is measured, usually, by the speed, accuracy, and the resulting output(s) of a student's response(s). Furthermore, performance infers the rate and quality of learning. Therefore, by closely monitoring the performance of the student through information stored during the simulation we can gain insight in the effectiveness of the learning process.

4. Learning Enhancement

Learning can be enhanced in a number of ways if the simulators inherent capabilities are properly organized and employed. It can standardize practice settings; systemize student exposure to tasks, subtasks, and task elements, and to conditions relating to the development and exercise of operator skills; it can be operated in fast or slow time; it can be stopped to permit an on-the-spot review of performance; and it can be used to replay the essential features of the student's performance for critique and for problem diagnosis and evaluation.

3.6.6.4. Types and Use of Training Simulators

Two types of operator training simulators can be distinguished: (i) the *basic principle* (BP) type, also called *compact* simulator, and (ii) the *full scope* (FS) simulator.

The BP simulators (often generic) are designed to present the integrated plant dynamics without all the details of a real plant. The instrumentation, considerably reduced compared to that in a control room, is well-arranged to allow for easy understanding. These simulators are generally used (examples can be found in: Gnosspelius, 1983; Begg and Weldon, 1983; Stanley, 1983) as part of the plant specific training, to enhance the understanding of the plant control principles and reactor dynamics, both in normal conditions and during a number of malfunctions.

The FS simulators provide the possibility to train operators in normal or accident situations under realistic circumstances. Each simulator is a close replica of a specific process unit, with a similar control room, and mathematical models of the plant's components -based on physical laws- programmed in computers well suited for real time applications. Practically all the instruments of the reference plant control room are represented (examples can be found in: Ancarani and Zanobetti, 1983; Gnosspelius, 1983; Lakey and Gibbs, 1983; McFarlane, 1983; Hamilton et al., 1983; Myerscough and Hignett, 1983, Walsh, 1983; Waterfield and Wyse, 1983; Lhoir, 1984; Tompsett, 1987; Leitch and Horne, 1988; Gran et al., 1988; Jones and Brook, 1990).

These simulators are designed to fulfil the basic requirement of realism: every event displayed in the simulator's control instrumentation shall be explicable on the basis of facts known about the simulated plant. Specific plant simulation was adopted in order to make possible the use of data and documents from the real plant as reference material for training.

The degree of realism necessary is a function of how simulation is to be used as a training aid. This depends greatly on the structure of the training programme. Two extreme categories of training simulators can thus be identified, depending on the degree of realism or fidelity required for training and assessment purposes: (i) high-fidelity simulators, and (ii) low-fidelity simulators. Baker and Marshall (1989) suggest three main dimensions of fidelity:

1. *Physical fidelity.* This relates to the realistic representation of the working environment. For example the realism of the simulated control room in terms of size, layout, arrangement of instruments, etc.
2. *Operational fidelity.* This concerns the representation of the actual process dynamics, cause and effect relationships etc., in view of process operation under a variety of conditions.
3. *Psychological fidelity.* This involves the extent of duplication of the psychological features of real operation. For example, task complexity, perceptual skills, decision-making and stress.

Different training objectives will require the use of different levels of each of the fidelity types. "For example, in the early stages of learning about operation, the physical fidelity of the training environment may not be of paramount importance, nor would it be advisable to include complex or stressful tasks. However, in the more advanced stages of training, and especially for assessment purposes, we can assume that a greater level of fidelity in all three dimensions will provoke more realism in the observed performance of trainee operators" (Baker and Marshall, 1989).

One should note that the degree of fidelity is not necessarily representative of the type of simulator. In other words, high-fidelity does not necessarily mean a full scope simulator (Bainbridge, 1990).

The structure most widely used in the process industries for incorporating simulators in the training programme is exhibited in figure 3.1. It is seen here that the operator is first presented with the theoretical background necessary for the understanding of the system. Then follows plant specific training in which BP simulators can be used. After this part of the training is completed then the operator can go on to the FS simulator training which deals with the operation of the whole system. Each of these steps have their own distinct structure which aims to achieve the training objectives. Figure 3.2. displays a typical structure of FS simulator training.

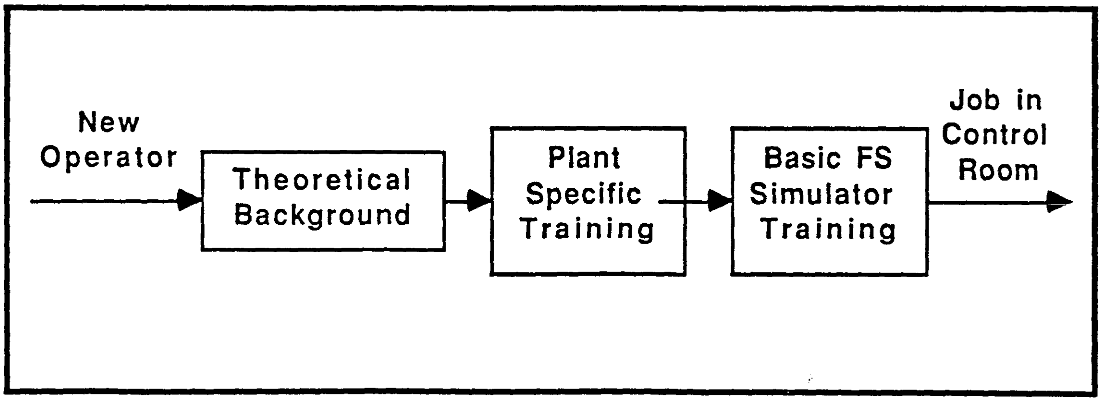


Figure 3.1. Structure of Training Program

The examples given above while representative are by no means the only possible or even effective ones. For instance, it is unlikely that a rigorous engineering quality simulator is needed for such training as console familiarisation or the basic skills of a control room operator.

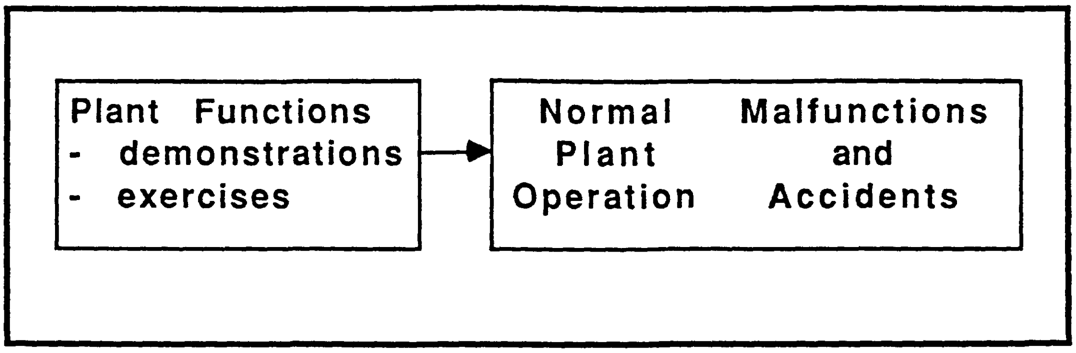


Figure 3.2. Basic Full Scale Simulator Course Structure

A common theme that is identified when the application of training simulators in the process industries is examined is that they are more widespread in the Nuclear Industry. This can be explained in view of the disastrous effects that may result due to a nuclear accident. For this reason the nuclear industry has tried to minimize the negative human effect by means of better training. This combined effort was undertaken primarily as a result of the Three Mile Island accident which was caused by an operator error and led to the re-evaluation of training practices.

The effort for the development of effective training packages was also greatly assisted by the fact that the issues facing nuclear power plants' operation are not as diverse as the issues faced in the chemical industries. This fact enables designers of training systems to enhance training concepts and improve the training facilities in view of a 'constant set of constraints'. Thus, the experience of the Nuclear Facilities in developing training systems represents a source of valuable information for a training system designer.

3.6.6.5. Representative Training Simulators

In this section some of the systems that exemplify the use of simulators for training[6] are presented. The examination of training simulators above has been based

---

[6] Two of the most advanced BP training simulators that have been implemented in an industrial environment are STEAMER and RBT. They provide some innovation over more conventional training simulators. However, due to their special characteristics arising from the use of ICAI techniques they were examined in section 3.5.6.5.4. of this thesis.

on existing systems, and as such it serves as a presentation of the various characteristics and functions of the state of the industry.

1. PCTRAN

Li-Chi Cliff Po (1988) describes a BP training simulator aimed at the Nuclear Industries, PCTRAN (Personal Computer Transient Analyser). PCTRAN has been developed as a supplementary training tool for the training of operators and engineers. Its main function is to allow for emphasis to be placed on the learning objectives. Therefore, PCTRAN has been used both for preliminary training of junior people prior to their formal full scale simulators and for re-qualification of experienced personnel concentrating on specific events.

Available for the IBM PC PCTRAN takes "full advantage of the PC' s interactive capability and colour graphics display." Generic software for both PWRS and BWRS are available with models representing core kinetics, coolant flows, feedwater and steam control, emergency core cooling systems and containment structure. Plant specific parameters are constructed as a data set and are used as input to the code.

The execution is menu-driven. The user is directed to select an initiating event ranging from an operational transient such as a valve or pump malfunction to accidents involving pipe breaks. "During execution a high-resolution colour mimic is displayed on the computer monitor. Selected plant parameters are digitally updated and the water level, relief valve status, pump rotors and control rod movements are analog displayed".

2. A Training Simulator for a Low Pressure Distillation Column.

This BP simulator was developed by HENKEL for the inhouse training of process operators (Luder et al., 1990). The focus here is not on the operation of a practical process with a specific control structure under special conditions, which would be the case in the safety training or in the training of start-up and shut-down strategies for experienced personnel. Instead it aims to demonstrate to the operators the fundamentals of process operation including: (i) use of the displays of a distributed control system (DCS) to observe the process, (ii) practice the operation of control valves, (iii) learn to collect, select and interpret relevant information, and (iv) develop a problem-solving strategy that only needs to be verified by observation and does not depend on short time validation measurements unavailable due to large time constants.

The training simulator consists of a dynamic process model which is linked to the actual DCS which serves as the interface for observations and control actions. The main display is a flowsheet of the column including the measurements of flows, levels, temperatures, pressure, and the concentrations of the key components is displayed at the DCS. It also includes information not available in the real plant but which are thought to enhance the learning process.

The control configuration is not the most desirable for operation but it is regarded as a good selection for education. It requires a lot of operator action and the temperature controllers force the trainee to use his process knowledge concerning the pressure-concentration-temperature coherence. The main function of the simulator is that it allows the instructor to select predefined scenarios of training sessions, and choose among various disturbances and changes of operation.

3. A Training Simulator for Studies in Process Control

Various training programmes for introducing undergraduates studying Chemical Engineering to process control have been developed and implemented using IBM's RTPMS (renamed from ACS) system. Descriptions of this can be found in Kershenbaum and Macchietto (1984), Koppel and Sullivan (1986) and Postlethwaite (1987). The use of the simulator follows a formal course on process control. The objectives of the simulation based project include the design of control strategies and the implementation of the control theory in practice, in a realistic real-time environment. The students practice tuning control loops, study time constants as well as interactively operating a 'plant'.

RTPMS provides the platform for both the development and execution of this process model. Furthermore RTPMS is used as the operator/engineer interface to the process. The students are presented with a furnace simulation and are given the operational constraints. They then have to design and implement an effective control strategy that satisfies the operational requirements and is 'robust' in the presence of disturbances. "After investigations of the dynamic response to manipulations and disturbances, the students go on to look at the simple, separate, feedback loops to control the product outlet temperature and the stack oxygen concentration. The last unit looks at the effect on control of operating loops simultaneously and points out the difficulties in controlling multiple-input multiple-output processes" (Postlethwaite, 1987).

The course very effectively gives the students a better feel of the theory while at the same time it exposes them to some of the tasks that are associated with the role of the process operator in an actual plant. Koppel and Sullivan indicate that the "students when operating the simulated process from the ACS display experience virtually the same events that would confront operators of the real process, with the exception of drastic events such as equipment failure[7]."

4. Oseberg Platform Training Simulator

The Oseberg full-scope high fidelity training simulator (Gran et al., 1988) has been built and commisioned in 1987 with the primary objective of training process operators and supervisors for operating the plant under normal and abnormal conditions. The simulator (approximate cost of U.S. $ 6 million) is "one of the most comprehensive process simulators built to date" (Gran et al., 1988).

---

[7] Equipment failures can be simulated in RTPMS but they are not included in this course. For an example of this see Kassianides and Macchietto (1990).

The simulator includes a full replica of the platform's central control room (CCR) including a thorough emulation of the operating system with all the mimic and safety panels in the CCR of the platform reproduced in the simulator. The CCR panels consist of the process mimic panel, ESD matrix panel and the fire and gas matrix panel. Finally the simulator is backed by a comprehensive instructor system with dedicated functions for the running and control of training sessions.

More than one hundred main plant items were modelled in the Oseberg simulator. For example, crude inlet heaters, three phase separators, coolers, scrubbers, flash drums etc. In addition to the main items hundreds of transmitters, switches, alarms, controllers and valves are modelled. Other than equipment modelling, the simulator treats in detail the compressor sequencing logic system, as well as, the emergency shut-down and the process shut-down logic systems.

5. Training Simulator for an Olefins Plant

A replica, high-fidelity training simulator (introduced in section 3.6.1.) was developed by Simcon U.K., a division of Combustion Engineering for Shell Nederland Chemie's Lower Olefins plant at Moerdijk (Jones and Brook, 1990). The training objectives that were addressed by the training simulator were: (i) operator familiarisation of a newly installed distributed control system (DCS), (ii) operator training on DCS and de-bottlenecked plant interactions, (iii) training of new replacement operators, (iv) refresher training courses for all operators on a continuing basis.

For this purpose the simulator provided an accurate real time representation of the process dynamics. While the main components were simulated some secondary equipment were also included in the simulation models, that were identified as important for training. Furthermore, the need for the same type of interface through which the operators would monitor and control the re-vamped plant was provided by an accurate emulation of the actual control system.

While trainee operators are monitoring and controlling the operation of the simulated process model, instructors act as field operators to perform actions at the request of the trainee operator which cannot be completed in the control room. For example, opening and closing block valves, starting pumps and reporting readings from field instruments to verify the accuracy of control room instrumentation.

Instructors can also introduce disturbances, failures and malfunctions during training sessions which affect the operation of the process model. For example, "drift" failures were allowed for every controller transmitter, air failure options were included for every control valve, pumps and compressors can fail etc.

3.7. Discussion

In this chapter the area of training as it applies to the process operator was examined in some detail. It was pointed out that the effectiveness of a training programme depends greatly on a series of pre-training steps that have a direct implication on the training design and the choice of the most appropriate training method and training delivery platform. These pre-training steps have been defined as: (i) the identification of the trainee characteristics, (ii) the definition of the training needs, and (iii) the formulation of the training objectives through an extensive task analysis.

Various training methods that are currently employed for the training of process operators were then examined, indicating their respective advantages and disadvantages. As a result training situations were identified that stand to benefit from the application of appropriate CBT. Particular benefits are expected in the training of novice operators in plant operation and in maintaining operators' abilities to handle interactive real time functions.

Also in this chapter a detailed survey of CBT was presented which examined its cost-effectiveness and then looked in detail at the various CBT categories and how they relate to the process industries' training requirements. The findings of this survey indicate the following:

1. CBT where applicable offers considerable advantages over other training methods.
2. A great range of CBT courseware has been implemented, ranging from simple basic principle tutors to sophisticated high fidelity full scope training simulators.
3. Currently CBT is generally used to supplement the principal instructional process.
4. There is a need for an expert instructor to guide trainees and support the training process.
5. The routine use of CBT has been inhibited by the costs involved in the initial development of CBT material.
6. Subsequent maintenance and modifications of CBT courseware are often difficult and costly to implement as they are implemented in custom-made hardware and software systems. This leads to the rapid obsolescence of the entire training system.

Furthermore the survey indicated a number of broad requirements as defined by the process industries training needs which provide the framework and define the constraints for the development of CBT systems. These systems should be:

1. Stand-alone so that they could be used on the factory floor
2. Flexible and easily modifiable to account for and accommodate the changing nature of the operators' tasks and changes in the training objectives.
3. Capable of providing on-line criticism and support to enhance hands-on training.
4. Developed and maintained at reduced overall costs

In particular, a number of broad functionalities were identified as essential for flexible, easy to use and easy to upgrade CBT systems. With respect to CAI (or ICAI) systems, these are:

1. The ability to develop comprehensive well defined domain models that are accessible for subsequent modification.

2. The ability to employ both qualitative and quantitative knowledge representations of the domain.
3. The ability to express the domain knowledge through various delivery methods including text, diagrams, interactive displays, etc.
4. The ability to employ different training strategies at different times and for different training objectives.
5. The ability to select and save parts or all of the training session for either the on-line or subsequent evaluation of the student's performance.
6. The ability to use the above material in different ways to create stand-alone adaptive CBT systems that will cater for the needs of individual students.
7. The ability to manipulate the domain model and the instructional strategies to create systems that meet a range of training objectives.
8. The ability to expand the domain model, to implement new instructional strategies, and to implement new student models on the basis of experience and research.

With respect to simulation based training systems, the functionalities required are:

1. The ability to model a plant in a high level engineering language. To minimize development effort, it should be easy to develop new models, to reuse models already developed for design, control, or other purposes; to build complex models by assembling simple ones; to use a model library.
2. The ability to solve steady state, dynamic simulation and optimization problems with the same model, and for different and arbitrary initial conditions; to save and reuse intermediate solutions; to flexibly change specifications for the same model;
3. The ability to set up and run simulations from an external driver, including a control system; to run simulations syncronously with an externally defined time (including faster and slower than real time).
4. The ability to define, save, schedule and initiate procedural events and sequences through a high level language, independently and externally from the underlying simulation; to build complex procedures by assembling simple ones; to use a procedure library; to drive a simulation from the procedure definitions; to drive a control system from the procedure definitions.
5. The ability to interact with and use all facilities of a modern real time control system with support for multiple users and external links; to define its configuration from the outside; to access and use real time data.

6. The ability to define, store, and initiate, in a coordinated way, activities involving all the previous functions, according to predefined logical structures; to use the same material in different order; to mix procedural and logical data and programs.
7. The ability to communicate with trainees and trainers through suitably rich interfaces, including multiple screens, graphics, interactive dialogues, etc.

Finally, it has been identified that the integration of CAI (or ICAI) with training simulation in one training programme could offer great advantages. Furthermore, it is thought that the present state of technology could enable the development of systems that meet the majority of the forementioned functionalities. In the next chapter a system architecture that meets the training requirements for process operators and offers these functionalities is examined.

Chapter 4

AN INTEGRATED ARCHITECTURE FOR COMPUTER BASED TRAINING SYSTEMS

In chapter 3 a survey indicated various essential functionalities of flexible and cost-effective CBT systems. The system architecture should enable the development of CBT courseware for both basic principles and 'hands-on' plant operation whilst minimizing the development effort and the subsequent involvement of instructors. This could lead to substantial cost reductions while increasing training effectiveness.

In order for the system architecture to offer the required functionalities it must include a variety of diverse but highly sophisticated subsystems for simulation, real-time control, CAI authoring, etc. It was thought unlikely that any one organisation could provide all those components, and the skill base to support and develop them, at a cost sufficiently low to make the results widely available. Accordingly, we explored the feasibility of using, as far as possible, existing specially suited general purpose components as building blocks. In this chapter the overall system architecture and its current prototype implementation is described, in terms of the constituent components and their functions, and the links between the different modules. Then various CBT configurations that have been achieved within this architecture are presented.

4.1. The System Architecture

The proposed integrated system architecture is presented in Figure 4.1. It utilises high level general purpose modules: an engineering orientated process simulator, an industrial real-time plant management and control system, a supervisory system for scheduling and management of discrete operations and an intelligent module (implemented in an expert system development environment) which coordinates the others and serves as the platform for the development and delivery of the CAI courseware.

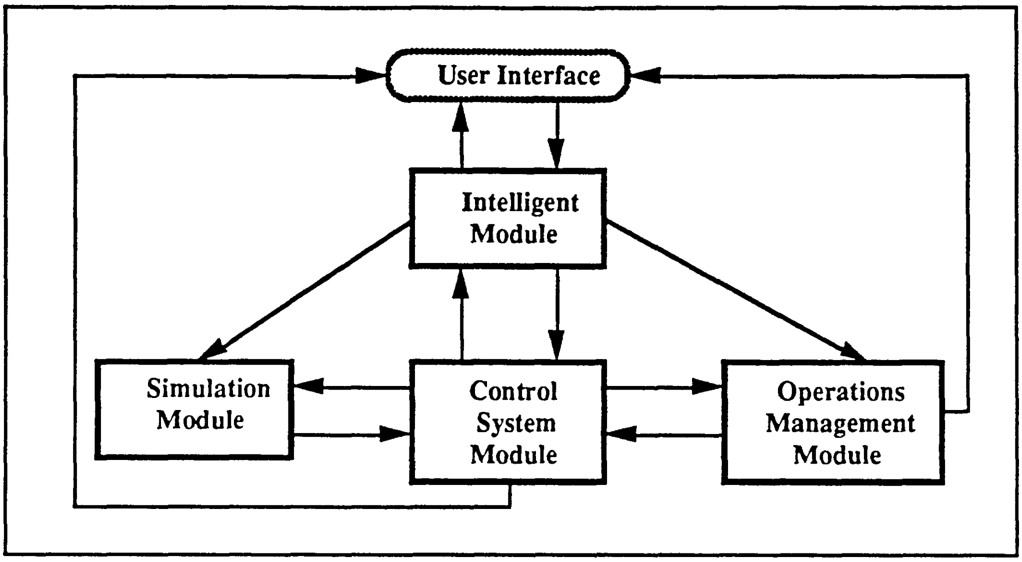


Figure 4.1. Structure of the Integrated System Architecture

The modular approach allows for the implementation of 'a generic environment' that can easily be adapted to different problem domains. Once the communication between the various modules is established then the effort involved in the adaptation of this generic environment towards a specific training programme is minimized. This is achieved through the exploitation of the functionalities offered by each of the building blocks.

For example, the trainer can develop the required simulation models in an environment tailored for such development, implement the control strategies in an appropriate medium, etc. Arguably this will require the courseware developer to be familiar with the workings of each building block. However, once that is achieved, modifications to the current problem domain, as well as development of new training programmes, maybe for different problem domains, should demand considerably less effort.

The system architecture is not dependent upon the use of specific commercial software as subsystems. However its sophistication is directly related to the functionalities offered by each of the building blocks. A set of constraints which determine whether individual blocks are suitable for use as modules in the integrated system have been identified and are examined more closely in the following sections.

4.1.1. The Control System Module

The control system module is of direct relevance to computer based training simulators. In chapter 3 it was indicated that in order to support learning at both the intellectual and the psycho-motor levels, a training simulator needs to incorporate controls and displays that are as real as possible. A variety of 'control systems' (ranging from real industrial systems to control system emulators of varying complexity) could be used for training purposes. The suitability of any one in particular will depend on the end use environment, the training objectives, the fidelity requirements and also upon cost considerations. The control system module needs to meet four functional requirements (listed in table 4.1.) in order to be eligible for use as a building block within the proposed system architecture.

Table 4.1. Functional Requirements of the Control System Module

1. It should allow the development and implementation of realistic user interfaces.
2. It should allow the interactive real-time execution of control actions.
3. It should allow the interactive (re)configuration of control strategies.
4. It should be able to communicate with external modules.

In the case of simulation training for plant operation the control system module will provide the point of interaction between the student and the simulated process plant. Therefore, the module should allow the development and implementation of appropriate user-interfaces, that include at least the same (or similar) features as those of the control system that the trainee will use to operate the real plant. This involves both the presentation of process data at realistic time intervals through representative interactive displays, graphics, etc. and also the emulation of the psycho-motor characteristics of the man-machine interaction.

The second functional requirement results from the fact that the trainee will control the simulated plant through these interfaces. Therefore, the control system module should allow the interactive real time execution of all possible (or the most important) control actions that the trainee and the trainer could perform with the real control system (for example, setting-up control loops, changing setpoints, etc.).

Additionally, manual actions (such as operating pumps, opening and closing manual valves, etc.) not available from the real control system configuration could be described in order to provide a comprehensive training platform. Furthermore, the control system module should provide for tutor intervention (direct or indirect) so that the trainer may initiate training scenarios such as control system faults, changes in the operational conditions, etc.

The third functional requirement adds to the flexibility as well as augments the possible training applications of the system architecture. The control system module should permit the configuration of alternative control strategies. This constraint guaranties the feasibility of maintaining and modifying the 'instructional control system' so that it corresponds to the current state of the real one. It should be possible to change control configurations using the available plant components, as well as implement new control loops.

When used by the trainer this functionality offers great flexibility for the definition of the training environment. Furthermore, this functionality also permits the development of training programmes aimed at teaching aspects of process control. In such cases, trainees could be allowed access to this facility to implement and test their own or suggested control strategies in a realistic environment.

Finally, the control system module should support communications with external modules. In the proposed system architecture the control system module is a key link in the transfer of data and the channelling of procedural actions between the other modules. A link which allows the sharing of process data with the simulator is necessary for the implementation of a training simulator.

Through this link the control system will drive the simulation as if it were an actual plant. The decoupling of the control and simulation modules enables one to easily manipulate the relative speed of the simulation and the control actions, thus augmenting the simulator training with 'slow-motion' or 'fast-forward' functionalities. Furthermore, it could allow the 'user' to initiate, freeze, pause, or restart a simulation problem.

On the other hand the links with the operations management and the intelligent modules enable these external programs to direct the operation of the control system - and through it, of the simulation- by passing to it data, parameters, values to trigger other actions, etc. As a result the resulting CBT program can have stand alone capabilities since training scenarios can be instigated automatically. These functionalities will be examined in more detail in the following sections.

In the prototype implementation of the proposed system architecture the control system module utilises an industrial system which meets all four functional requirements described above and also offers additional functionalities (IBM's RTPMS, Bradbury and Brook, 1983; *Process Engineering*, 1985). RTPMS is used in our department to operate three pilot plants and a variety of simulations. It provides standard real-time monitoring and control functions, along with alarming, archiving, event logging, and reporting capabilities. It has a powerful real-time database, a full operator interface and it supports multiple user access through configurable interfaces. In addition, RTPMS permits the interactive creation of simple and multi-loop control strategies, as well as the development of sequence control.

RTPMS can control access to different functions and data according to user type (operator, engineer), terminal location and authorisation. It can also initiate various external programs (when triggered by events, changes in continuous variables, time, etc.) and exchange data with them, as well as be directed by them to perform various functions. The communication between these and RTPMS is performed through suitable database server programs (Macchietto et al., 1987). Furthermore, RTPMS offers limited simulation facilities which can be used to simulate and control a process within itself. Table 4.2. outlines RTPMS's functionalities.

Table 4.2. Characteristics and Functionalities of RTPMS

1. Real-time monitoring.
2. Control capabilities.
3. Permits the creation of simple and advanced control strategies.
4. Permits the development of sequence control
5. Alarming and reporting capabilities.
6. Own real-time database.
7. A full operator interface
8. Multiple user access.
9. Archiving and event logging facilities.
10. Standard simulation facilities
11. Controls access to functions according to user authorization
12. Supports a two way communication with external modules

The use of RTPMS in the training system offers four main advantages. First, it makes it possible to present the trainees with interfaces identical to those they will have to use when operating the actual plant. Second, there is no need to emulate the control system itself since the real thing can be used. Third, the open architecture of RTPMS allows it to play a central role in the communications structure of the overall system architecture. Finally, RTPMS's functionalities increase the flexibility and the generic nature of the system architecture as a platform for the development of CBT systems.

4.1.2. The Simulation Module

This module provides the foundation for computer based simulator training. While the control system module is generally responsible for the interaction of the student with the training simulator, the simulation module provides for the development and execution of plant models.

The use of general purpose process simulators which are specially suited for the development and execution of the simulation models to overcome current limitations in training simulators was discussed in section 3.6.6.2. Simply, once the simulation models have been developed they could be linked to the control system module to achieve flexible training simulators. A process simulator that could be employed as the simulation module in the integrated system architecture needs to meet several functional requirements (listed in table 4.3.).

Table 4.3. Functional Requirements of the Simulation Module

1. It should have a generalized high level (engineering) definition language for ease of model development and subsequent modification.
2. It should allow the development of models that have a degree of fidelity which corresponds to the training needs.
3. It should be capable of solving steady state, and dynamic simulation problems with the same model and for different and arbitrary initial conditions.
4. It should be able to communicate and share information with peripheral modules.
5. It should be capable of running simulations synchronously with an externally defined time (including faster and slower than real time).
6. It should be possible to initialize, begin and control the execution of the simulation model from an external source.
7. It should be possible to freeze, restart and re-run a simulation.

The availability of a generalized high level (engineering) language should greatly reduce the effort involved in the development of new models and their subsequent modifications. Furthermore, this should provide consistency in the definition of models and enable the formulation of an extensive model library (including both complex and simple models). Thereafter, the various existing library models could be used to easily build new ones of higher complexity, and so on. Finally, models which are or were developed using the simulator's language for design, control, or other purposes could easily be modified and adapted for use in training situations.

It has been repeatedly stated that the degree of fidelity necessary in a training simulator depends greatly on the training objectives. In view of the implementation of a 'generic' system architecture which could be used to address a range of instructional objectives, the simulation module should allow the description and be capable of handling both discrete and continuous events.

One of the greatest cost factors in the development of training simulators has been the implementation of robust numerical routines that are able to solve the simulation models. This demand increases significantly if the plant models are large and include a combination of both discrete and continuous events leading to large mathematical systems of algebraic and differential equations. The simulator should be capable of handling such models. Furthermore, it should have the ability to solve steady state, and dynamic simulation problems with the same model and for different and arbitrary initial conditions.

This would allow the use of the simulation module to address a range of training objectives ranging from the presentation of direct cause and effect relationships to the introduction of time considerations in plant operation. Finally, the simulation module should offer the capability of saving and using intermediate solutions, as well as offering the flexibility to change specifications for the same model. This functionality further increases the applicability of the simulation models to different training scenarios.

For a simulator to be suitable for use as the simulation module in the proposed system architecture it needs to be capable of communicating and sharing information with peripheral modules. Firstly, a link with the control system module is essential for the configuration of even the simplest training simulator. The need for this link should be evident and has also been discussed in previous sections of the thesis. The main constraint imposed upon this link is that the simulator should be capable of running simulations synchronously with an externally defined time (including faster and slower than real time) in this case the time defined by the control system module so that the dynamic effects of control actions upon the plant operation are realistically represented.

Furthermore, the ability to select, initialize, begin and control the execution of a simulation problem from an external source, increases the suitability of the simulator for use as the simulation module. Once the models have been developed and tested then they need not be directly presented to either the trainer or the student. The execution of the simulation models should take place in the background. Executable versions of the simulation models could be used to formulate a 'simulation problems library'. Individual simulation models may then be defined as parts of an overall domain model at different levels of a CBT programme. As such, the trainer, the student, or a predefined automated training programme could select any of the predefined problems for study.

Once the selection of a problem is made, then it should be possible to initialize and run the respective simulation automatically, thus removing any requirements on the part of the student for being familiar with the use of the simulator. Furthermore, the ability to control the execution of the simulation from external sources allows the training developer to define various simulation procedures. These procedures could then be invoked either directly by the trainer and/or the student through specially suited interfaces or be utilized automatically from within an automated training programme.

Finally the ability to freeze, restart and re-run the execution of simulation models allows for emphasis to be placed on important events and sequences. By freezing the simulation the student can take time to study the characteristics of an event as described by the state variables at that particular instance, as well as seek help in the form of either explanations or advice. This help could then be made available by either a human tutor supervising the instructional process or automatically from a facility offered by a CBT environment.

Additionally in the case of a stand-alone system used for self-study, the student may pause the simulation to take a break or perform some other non-training function. If for example an operator is using a training system in the control room his attention may be required for actual plant operation. In that event he may freeze the simulation, perform the non-training operational task required and then return to the training system and resume with the training session. Alternatively the trainer may wish to freeze the simulation in order to describe a particular event, offer additional information, or even test a student's understanding of what is happening.

The ability to restart the simulation provides for continuity of the instructional process. For the effective use of the freezing functionality described above it is necessary that the student is able to resume with the training session. At a different level the student may wish to restart a simulation problem at a particular point in order to repeat the training process. If for example the objective of an exercise is to devise a sequence of procedures which achieve certain operational objectives, the student may interactively examine various alternatives until he makes his final choice.

Finally, the ability to re-run a simulation problem with all the actions that the operator has taken during an interactive session, allows the instructor, as well as the student to critically examine a student's performance and the effectiveness of his decisions in view of the final results.

In the implementation of the proposed system architecture the simulation module makes use of the SpeedUp simulator (Pantelides, 1988) which meets the previously described functionalities. In particular SpeedUp offers a high level simulation language and model development environment, and robust numerical techniques for the solution of the underlying algebraic and differential equations. Table 4.4. outlines the functionalities offered by SpeedUp.

Table 4.4. Characteristics and Functionalities of the SpeedUp Simulator

1. High level simulation language.
2. Model development environment.
3. Robust numerical techniques.
4. Communication with external modules.
5. Easy development of real-time simulation models.
6. External initialization at different dynamic states.
7. Support freeze, restart and re-run.
8. Can run in slow motion or fast forward.

A simulation model is first developed off line using SpeedUp as a stand alone program, and it is then interfaced to the control system module so that the simulation may be run in real time. SpeedUp's External Data Interface (EDI) and RTPMS's database server make the communication possible. The relation between the simulation module and the control system module is schematically given in figure 4.2.

The interface permits the SpeedUp simulation to receive signals such as valve positions from RTPMS, and allows the simulation to return calculated values such as flowrates, temperatures and compositions back to RTPMS at regular time intervals (every 8 seconds). While possible we normally do not include control loops in a SpeedUp plant model, instead we use the standard RTPMS facilities to configure control strategies and perform control actions enabling exact replicas to be built of the actual or expected plant control environment.

Furthermore, the interface allows the operator or the trainer to freeze, restart, or rerun a simulation through commands from the control system itself. An additional facility available is the ability to slow down the simulation or make it run faster by manipulating the apparent simulation time relative to clock time.

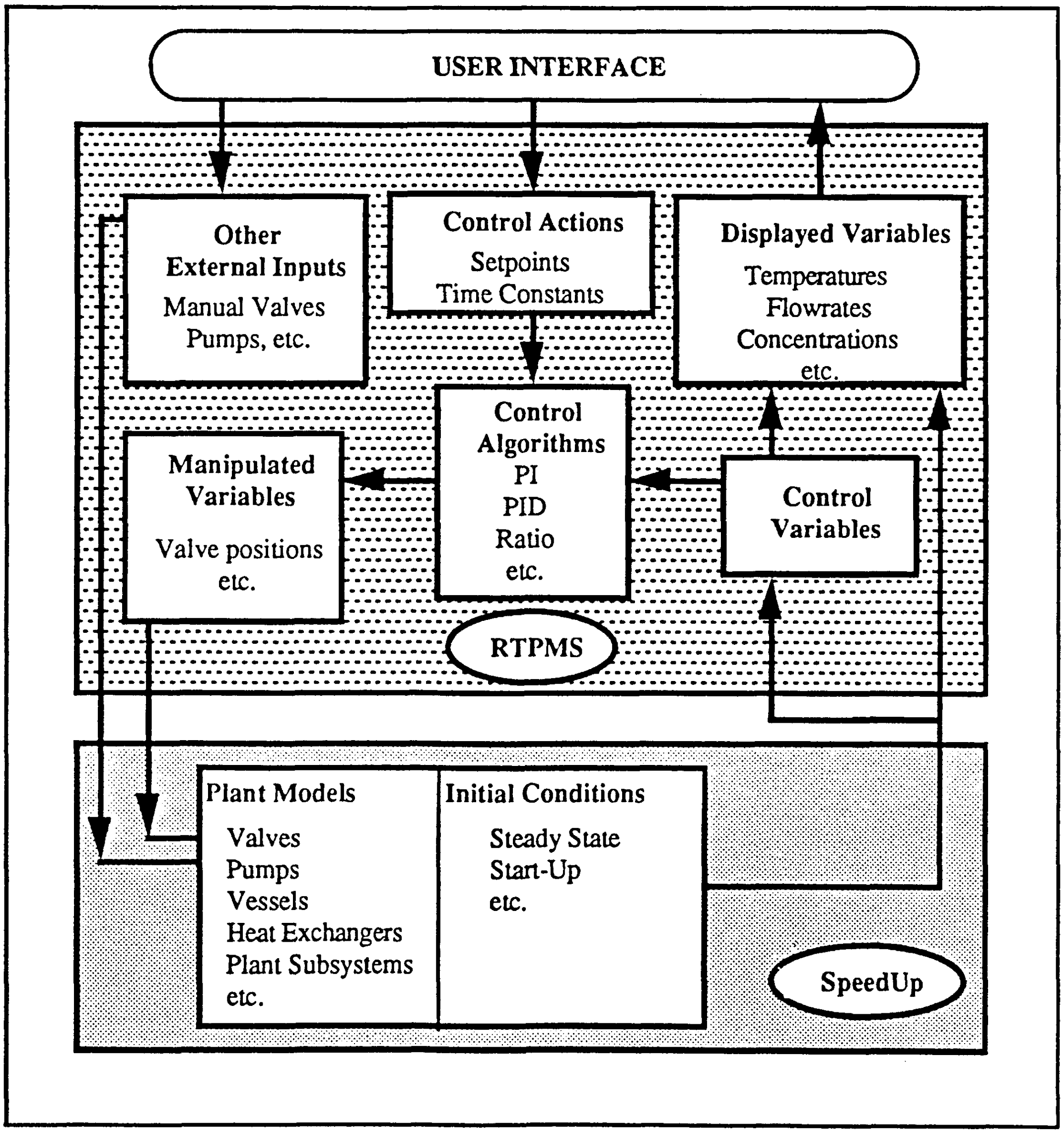


Figure 4.2. Relation between the control system and the simulation modules

A plant model is created by defining models of individual processing items, such as valves, tanks, heat exchangers, etc., and by adding purpose built models to those in the library. It is then possible to easily define simulations of the plant or of plant subsystems involving various combinations of these building blocks and solve steady-state, dynamic simulations and optimization problems.

Faults such as loss of containment through leakage, blockages, measurement shifts e.t.c. can also be incorporated into the simulation models. For this a plant model is coupled with error models which are also manipulated through the control system module. Error models can be used to affect sets of state variables or individual ones. Required switches and/or parameters in the error models can be provided by the trainer on-line or be automatically set/reset at a pre-specified time and/or conditions. Figure 4.3. shows how these error models are related to the main configuration exhibited in figure 4.2., for a specific example.

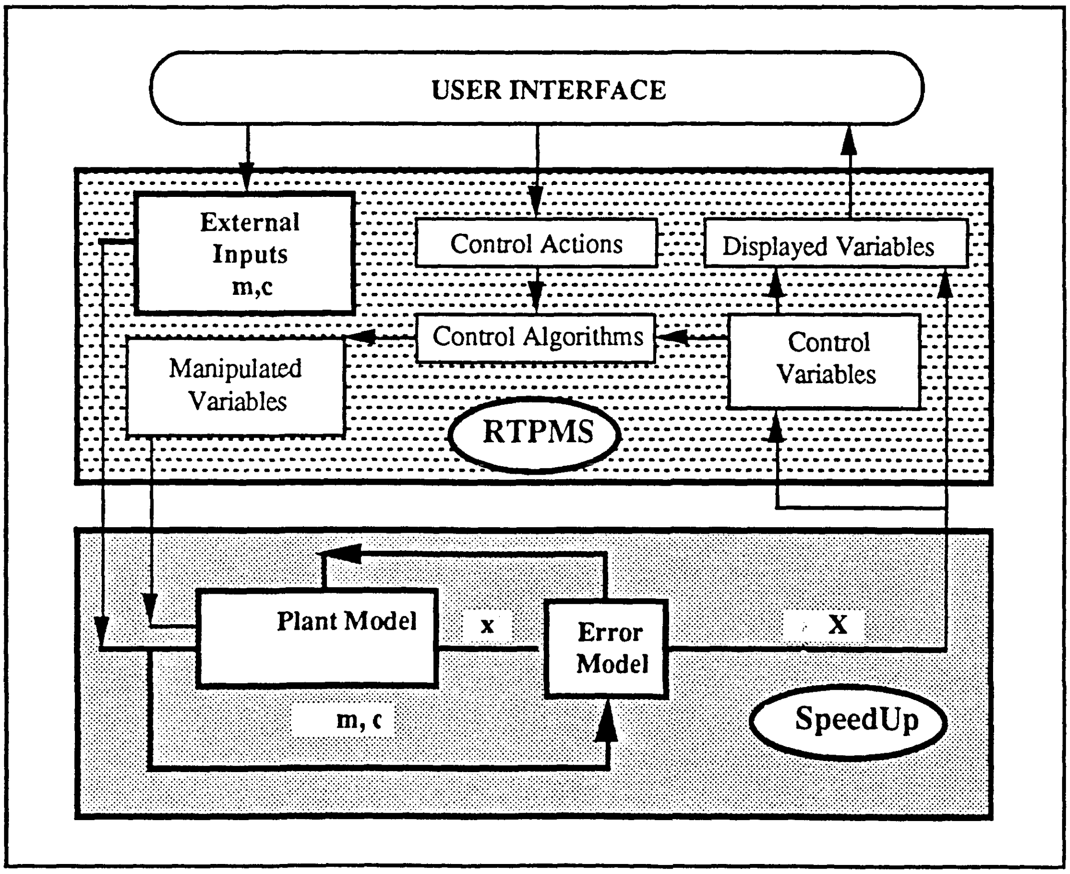


Figure 4.3. Incorporation of Error Models in the Simulation Module

A simple measurement error model and its effect are described in table 4.5. Let $\mathbf{x}$ represent a measured flowrate. If $\mathbf{m = 0}$ and $\mathbf{c = 0}$ then the value returned by the simulation to RTPMS is $\mathbf{X = 0}$. In the actual plant this could correspond to total measurement loss which could be explained by a non-functional measuring device, or to a complete loss of flow through leaking. If $\mathbf{m = 0}$ but $\mathbf{c \neq 0}$ then $\mathbf{X = c}$ and a constant measurement is reported to the control system. This could indicate a faulty instrument or it could correspond to a control valve that has become stuck in one particular position.

Table 4.5. An example of a simple error model

$$X = m^*x + c$$

Where:

X is the value sent to RTPMS
x is the value evaluated by SpeedUp
m is a constant transmitted from RTPMS
c is a constant transmitted form RTPMS

Effects of constants on values sent to RTPMS:

$m = c = 0$ total measurement loss
$m = 0, c \neq 0$ arbitrary constant value
$m = 1, c \neq 0$ constant measurement offset
$m \neq 1, c = 0$ proportional measurement offset and or leak
$m \neq 1, c \neq 0$ constant plus proportional offset

In the situation where $\mathbf{m = 1}$ and $\mathbf{c \neq 0}$ a constant measurement offset would be observed. This could correspond to a wrongly calibrated instrument, a constant leak, etc. If $\mathbf{m \neq 1}$ and $\mathbf{c = 0}$ then the value exhibited by RTPMS would indicate a proportional measurement offset possibly caused by a faulty instrument, leakage, blockage, etc. Finally if $\mathbf{m \neq 1}$ and $\mathbf{c \neq 0}$ then the effect on $\mathbf{X}$ would be a combination of constant and proportional offsets possibly explained by a combination of faults.

Furthermore, the effects caused by this simple error model can be grouped in two categories. The first category includes the effects upon isolated and/or decoupled variables. In this case the error model affects the particular state variable but does not have an effect on any other part of the simulated process.

The second category includes the effects upon variables that are used in a control loop or elsewhere in the simulation. In this case the offset caused by the error model affects different parts of the simulated process. Thus, the resulting effects are not isolated. If for example the measured flowrate is coupled to a flow controller then it is possible that the control loop will be destabilized. This could then have various effects downstream, maybe leading to operational instability.

Various error models may be defined and utilized in a training scenario, for example as a demonstration of the effects of a particular fault on plant operation or in an interactive session where the operator is guided by the instructional process to identify the fault.

In summary, it is possible to easily develop real time simulation models which can be initialized at different dynamic states, support freeze, restart and re-run, can be run in 'slow motion' or 'fast forward', and whose level of detail can be such as to reflect the effects of power switches, manual and computer operated valves, the setting-up of control loops, etc. A range of faults may be simulated, for example the loss of containment through leakage, blockages and equipment malfunction, and simulation of measurement and control system errors. Of course the fidelity of a simulation depends on the suitability and accuracy of the models used, but here it is possible to very rapidly develop and change a model in the SpeedUp environment and then to link it again to the training system.

4.1.3. The Operations Management Module

The integration of the simulation and the control system modules provide a platform for the development of training simulators with advanced functionalities. However, the resulting training systems achieved through this configuration lack in terms of stand alone capabilities. While it is possible to describe some operational and training sequences in the simulation module as parts of a plant model itself, the effectiveness of this approach is limited for two reasons. First, these sequences will have to be embedded in the model, thus constraining its general applicability as a generic building block. Second, the definition of such sequences in this environment would require significant development effort.

These limitations are overcome by using an operations management module which is specially suited for the development and implementation of operational procedures. The use of such a module greatly enhances both the stand alone capabilities of the training systems and the general applicability of the development platform for simulation based training.

The operations management module should permit the definition of complex operational procedures in terms of a set of predefined control primitives independently and externally from the underlying simulation, then translate these procedures into control commands to be automatically scheduled and executed under the system's supervision. In view of integrating such a module within the proposed system architecture six functional requirements have been identified (listed in table 4.6.).

Table 4.6. Functional Requirements of the Operations Management Module

1. It should have a high level definition language for ease of definition and subsequent modification of operational procedures.
2. It should be capable of translating these procedures into control commands.
3. It should be able to schedule and execute the resulting control commands.
4. It should be able to communicate with external modules.
5. It should allow the selection of sets of procedural events and their initialization from external sources.
6. It should allow on-line interactive operation.

As in the case of the simulation module the availability of a high level language should greatly reduce the effort involved in the initial definition and the subsequent modification of procedural events and sequences. The generality in the definition of these procedures should permit the development of a procedure library. Thereafter, complex procedures could be built by assembling simple ones.

The operations management module will have to be able to translate these procedures into commands that can be used by external modules. In the proposed system this module is linked to the control system module. Therefore, it should be capable of translating these procedures into control commands that can be executed at the control system level.

Furthermore, the operations management module should be able to schedule and execute these control commands. This enables complete sequences of events to be scheduled at prespecified time intervals or at a pre-specified time. This allows the complete definition of operational demonstrations, the introduction of faults and disturbances at a prespecified instance in time, etc.

The requirement for communication with external modules should be evident. The actual execution of the procedural events will be performed by an external module. In the proposed architecture this is the control system module. Therefore, the operations management module should be able to drive the control system from the translated procedure definitions. Additionally, it should also be able to drive the simulation via the control system by the utilization of appropriate procedure definitions.

The ability to select sets of procedural events and to exploit the functionalities of the operations module from an external source adds a new dimension to the stand-alone capabilities of the integrated system. This is particularly relevant to demonstrations of operational sequences (for example start-up and shut-down procedures). A description of how this capability is utilized in the proposed architecture is given in the following section.

Finally, it should be possible to on-line and interactively control the functions of the operations module. This allows for the trainer to initiate sequences or schedule events to happen on-line or at a future time. Alternatively, the student can interactively initiate a sequence of events and then observe its overall effects.

In the prototype implementation of the system architecture we used the SUPERBATCH supervisory operations management system (Cott, 1989; Cott and Macchietto, 1989). It permits the definition of a set of primitives, which include equipment resources, utilities, and operational primitives (phases or tasks that can be carried out). These may well differ from (but must be consistent with) the basic models used for simulation or control purposes, and superimpose different views of the same underlying plant/control configuration. These new primitives may be used to define operational procedures in terms of the known resources and to produce an operations schedule.

The system can translate these schedules automatically into control commands and direct their execution under its own supervision by the control system module. Schedules are updated on line in response to plant (in this case, plant simulation) events, user requests to schedule new procedures, etc. The communication between SUPERBATCH and RTPMS is performed through the database server previously mentioned. Table 4.7. outlines the characteristics and functionalities of SUPERBATCH.

Different SUPERBATCH models may be defined for the same plant/control configuration, allowing a plant to be viewed and operated at different levels. In a detailed operations model, the available resources may be valves, motors, etc. and the available procedures may be to open and close valves, put a control loop on automatic at a user specified setpoint, etc. A much more aggregate model might describe the same plant as a reactor, a distillation column and a heat exchanger, and the available procedures as load reactor, start reactor, etc.

Table 4.7. Characteristics and Functionalities of SUPERBATCH

1. High level plant/procedure definition language.
2. Model development environment.
3. Automatic translation of procedures into control commands.
4. Automatic scheduling and execution of control commands.
5. Communication with external modules
6. External selection and initiation of models.
7. On-line interactive intervention

From a training perspective the operations management module can be used by the trainer to define training sequences and scenarios. Consideration of general training objectives indicated a number of ways to use this sequence management capability. First, the system can be used for demonstration purposes. For example, the trainer can prespecify an entire procedure or sequence of procedures for successful start-up or shut-down of the plant. This predefined procedure(s) can then be invoked as a demonstration of a correct, or safe operation. As part of the predefined sequence, explanation may also be presented to the trainee at selected points.

Second, the scheduling facilities of SUPERBATCH can be used to define and generate a variety of training procedures. These may include normal operation where, for example, a trainee is expected to take the plant at a certain point, and emergency training scenarios. Abnormal operational conditions may be defined by automatically introducing disturbances and/or faults (for example, by shutting a manual valve) as part of the sequences. In this situation the trainee operating the real-time simulation would have to identify that there has been a deviation from normal operation, locate the source of the error and take corrective action. Part of a procedure set up by the trainer may include the collection of certain statistics for evaluation purposes. The main advantage here is repeatability of the training scenarios in the sense that exactly the same training sequence is used for all trainees and at different times.

Finally, a trainee may define, execute and verify his/her own operating procedures in terms of control primitives available to him/her. The extent and detail to which events can be formally defined depends upon the level of detail of the simulation model and the generality of the selected control primitives. For any particular training activity these may be controlled by the trainer.

For example, control primitives and procedures made available to a trainee may implement very simple tasks, such as opening and closing valves, turning motors on or off, etc. Alternatively, the trainee may be asked to operate the plant using higher level primitives such as "perform start-up", etc. For example the SUPERBATCH model developed for the evaporator pilot plant application (defined in detail in later chapters) includes primitives and procedures for:

1. Initiating, pausing and aborting the simulation.
2. Carrying out all manual operations (the operation of power switches, manual valves, pumps, and an agitator).
3. Performing all control actions (closing a cascade, putting a control loop on automatic and setting the setpoint).
4. Individually adjusting drifts and offsets in all measurements and displayed values.

Any or selected subsets of the above procedures may then be interactively scheduled and executed in any desired sequence.

The above features enable the training system to be used in stand alone mode since all training procedures, for example demonstrations and introduction of faults can be automated. This permits the system to be used without the trainer being present and for self-study.

4.1.4. The Intelligent Module

The integration of the control system, the simulation and the operations management modules (from now on referred to as the simulation based training system, SBTS) allows for the configuration of flexible and cost-effective training simulators of advanced functionalities which may have various degrees of stand alone capabilities. In chapter 3 it was pointed out that the integration of computer-assisted instruction (or ICAI) with simulation techniques in a single training environment could offer great advantages. For example, CAI techniques could be used to provide on-line criticism and support hands-on training. This could greatly add to the stand-alone capabilities of a training programme.

It may be possible to define a stand-alone training programme which utilizes CAI techniques for introducing instructional material on theory, basic principles, etc. and then take advantage of the capability of configuring training simulations, to define simulation problems that demonstrate the theory and/or provide a platform for the application of theory to practice. Furthermore, the control of the instructional process could be entirely automated by: the definition of structured training sequences augmented with explanations, guidance, advice, etc.; the implementation of different instructional strategies; the definition of student models; etc. Thus it may be possible to develop adaptive training systems that will attend to the training needs of individual students.

As indicated in chapter 3 there have been various attempts to achieve training systems with these specifications. However, their success has been greatly constrained by the limited capabilities of these systems with respect to simulation. Nevertheless, a great range of commercial authoring systems that allow the definition of CAI courseware are available. Dean and Whitlock (1988) describe the ones most widely used. On the other hand authoring environments that support the development of ICAI systems are not yet available. Instead ICAI systems have been developed for specific problem domains and have their characteristics and functionalities deeply embedded in specific software implementations.

These limitations suggested the general architecture of the overall system presented here. It was thought that the integration of an intelligent module with the SBTS could overcome these limitations and allow the development of flexible and cost-effective training systems that combine the functionalities of CAI, ICAI and training simulation. As a result the intelligent module would have to be capable of performing two roles. The first role is to represent and define the delivery mode of the instructional material. The second role is to coordinate the use of the other modules within a particular training programme.

4.1.4.1. The Intelligent Module Development Environment

Different implementations of the intelligent module are possible depending upon the training needs. Nevertheless, its implementation through the use of expert system techniques is the most advantageous. This approach allows the implementation of flexible data structures that can be easily formulated, updated, modified, and eventually manipulated to develop a range of CBT courseware that address different training objectives. In this way it is possible to define domain, instructor and student models, as well as non-training related functional parts (such as a fault monitoring facility) and use them to create specific training programmes (one such application is described in later chapters).

The use of one of a variety of commercially available expert system development environments greatly reduces the effort involved in the development of the intelligent module. A number of functional requirements have been identified which have a direct implication upon the sophistication of the training systems that may be generated. These are outlined in table 4.8.

Table 4.8. Functional Requirements of the Intelligent Module Development Environment

1. The availability of a high level definition language for ease of development and subsequent modification of expert systems.
2. The ability to handle mathematical functions.
3. The ability to develop comprehensive well defined domain models that are accessible for subsequent modification.
4. The ability to employ both qualitative and quantitative knowledge representations of the domain.
5. The ability to express the domain knowledge through various delivery methods including text, diagrams, interactive displays, etc.
6. The ability to employ different training strategies at different times and for different training objectives.
7. The ability to select and save parts or all of the training session for either the on-line or subsequent evaluation of the student's performance.
8. The ability to use the above material in different ways to create stand-alone adaptive CBT systems that will cater for the needs of individual students.
9. The ability to manipulate the domain model and the instructional strategies to create systems that meet a range of training objectives.
10. The ability to expand the domain model, to implement new instructional strategies, and to implement new student models on the basis of experience and research.
11. The ability to communicate with external modules, share information and coordinate their activities.
12. The ability to develop and incorporate additional functional parts which may require extensive logic based representations and processing.

LEVEL5 was used (Information Builders Inc., 1987) in the prototype system implementation. LEVEL5 is an advanced development environment and delivery vehicle for expert systems and uses both backward chaining goal driven inference and forward chaining. In a backward chaining application LEVEL5 will pursue a specific hypothesis or goal, searching for the antecedents (pre-conditions) that would support that goal. In a forward chaining application it attempts to match contextual data, or information specific to a particular situation, to a pattern or template described by the rules of the knowledge base.

When building expert systems one specifies a goal or a hierarchy of goals which are then proven or disproven by a network of interdependent rules. LEVEL5 makes use of a simple yet versatile knowledge representation language called Production Rule Language (PRL) for the development of knowledge bases. In PRL knowledge is represented as IF...AND...OR...THEN...ELSE rules, which contain the factual information comprising the domain of the expert system. PRL also allows the user to specify procedural rules, the execution of which are not dependent on the satisfaction of antecedents (IF conditions).

Numeric data can be manipulated using boolean, arithmetic, or higher mathematic operations. Furthermore, a numeric confidence level can be assigned to each conclusion and antecedent. Knowledge bases have a variable threshold or minimum confidence acceptability level that can be adjusted as the knowledge base executes. Knowledge bases created by LEVEL5 are run in compiled form, and thus execute quickly and efficiently.

LEVEL5 knowledge bases may activate other LEVEL5 knowledge bases and communicate with them via a file of global facts or parameters that are shared among all knowledge bases activated for a particular application. In this way linked knowledge bases may communicate with one another dynamically and update global facts with the execution of each knowledge base. Finally, LEVEL5 can receive inputs from external sources as well as activate external programs directly from a knowledge base.

4.1.4.2. The Intelligent Module, Functional Definition

Within the intelligent module the *domain model* contains instructional material which does not require dynamic simulation. In the application discussed later this ranges from qualitative information (in a combination of text and diagrams) and simplified steady state simulations (describing cause and effect relationships at plant component level) to more complicated steady state simulations which focus on the interactions between plant subsystems.

Sophisticated *student models* could be implemented within our environment which will adapt the instructional process to the needs and progress of individual trainees. The development of an intelligent student model, however, was outside the scope of the present work. Nevertheless, the structures of simple student models have been implemented.

The *tutor model* controls how and in what order the domain concepts are introduced and monitors the user's progress in solving domain problems. In the application mentioned above different instructor models were used which employed one or a combination of three instructional strategies. The employed strategies are : the *Socratic method*, the *coaching method*, and a *structured model* under which the student is formally presented with domain concepts in a structured sequential manner.

While it is possible to implement simple steady state simulations in the intelligent module itself (in LEVEL5), it is necessary to use external modules for more complex steady state and dynamic simulations and for real time hands-on training. The use of the intelligent module as a controlling device permits the incorporation of real time dynamic simulations at appropriate points in an overall training programme domain model. Thus, the intelligent module needs to act both as a User Interface Management System and as a coordinator of the functional operation of each of the other modules.

It can instruct each of the other modules on what is appropriate at a specific point of the training process. Through the communication link with the control system module the intelligent module can receive data that describe the status of the plant, it can indicate which control configuration to set up and suitable initial parameters values, it can send desired setpoint values, set up control loops, open and/or close manual valves, it can pause or abort the simulation through the control system, it can instigate faults, and it can indicate which procedures to be executed by the operations management module. Also, it can invoke various simulation models, suitable initial conditions, and procedural primitives. Generally all the actions possible at the control system module level can be initiated and controlled from this higher level.

To the simulation module the intelligent module indicates which simulation model should be activated and the initialization point to be used (for example, a start-up from an empty plant, an operation from a specific steady state, etc.). To the operations management module it indicates which management model should be used (for example, one including a detailed "view" of the available plant resources, two operators, and a particular set of available procedures with which the plant must be operated).

A functional part that has been included in the intelligent module application, described in later chapters, is a simple fault monitoring facility which will process data received from the control system module in view of identifying faults and inferring the future performance of the plant's operation. This sub-module is able to recognise which fault out of a predefined set of possible faults has occurred. The implementation of this facility allows the development of CBT courseware for training in fault identification.

This sub-module takes as inputs the current value of important process variables and the status of selected plant components (for example, the service status of a control valve). These variables are then treated in simple fault identification algorithms which identify the plant subsystem where a problem has occurred and the nature of the problem. Once the problem has been identified a LEVEL5 knowledge base is initiated that includes a training session to help the student identify the problem and understand its effects on plant operation. This is possible, since the possible problems that may be invoked by a training programme are know a priori and therefore the appropriate training sessions can be developed.

In summary, using the LEVEL5 environment it is possible to develop and implement intelligent modules with comprehensive domain models that represent conceptual knowledge, procedural knowledge and heuristic knowledge. The environment also permits the definition of different student models and instructional strategies. Furthermore, the intelligent module can act as the controlling device to integrate the simulation, the control system, and the operations management modules in one training programme.

Finally a simplified fault diagnosis sub-module permits process monitoring and the identification of selected faults out of a predefined set. Last, it is worthwhile to note that all the techniques used are of general applicability. Table 4.9. outlines the characteristics and functionalities of the intelligent module.

Table 4.9. Characteristics and Functionalities of the Intelligent Module

1. An extensive domain model.
2. Implicit student models.
3. Feasibility of introduction of an intelligent student model.
4. A range of instructional strategies in an instructor model.
5. Controlling device for the integrated system.
6. Reporting facility can store information describing a student's session.
7. Fault diagnosis sub-module.

4.2. Training System Configurations

In the previous sections the characteristics and the functionalities of the various modules that make up the proposed integrated architecture for computer based training systems were examined. These functionalities can be exploited in a variety of ways within a training programme. In this section several possible training system configurations are examined and it is shown that advanced functionalities can be achieved while minimizing the development effort.

4.2.1. A Standard Training Simulator

Once the communication link between the simulation and the control system modules is established a 'standard training simulator' (see fig. 4.4.) is achieved.

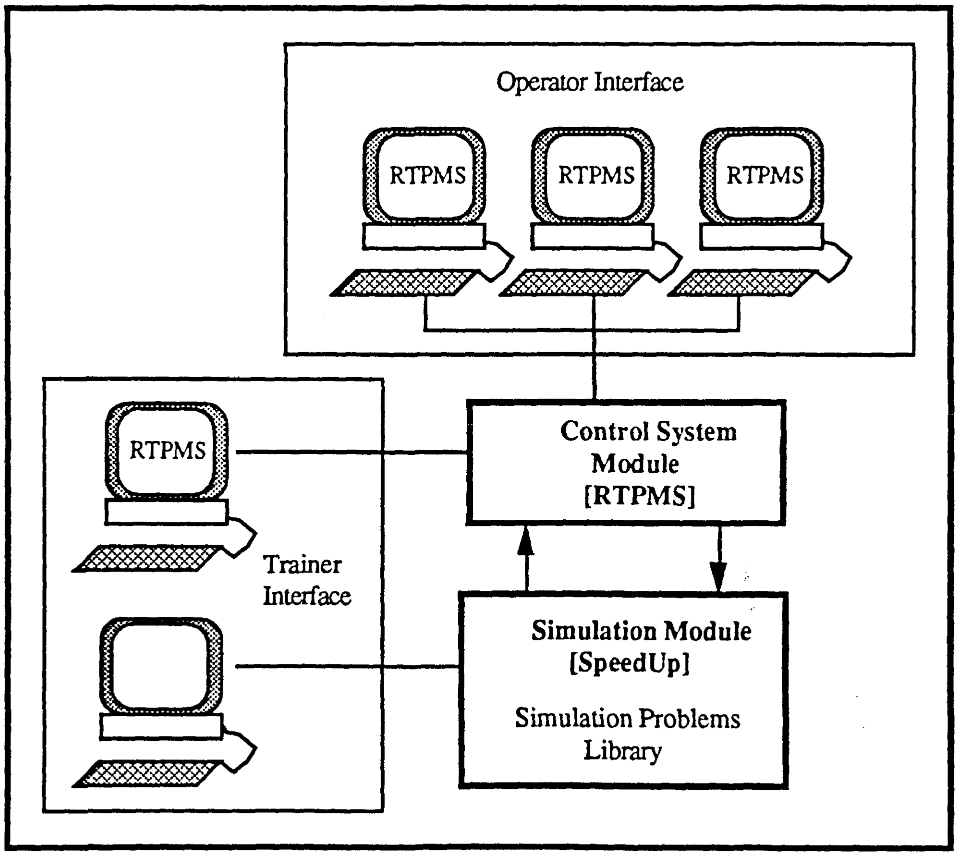


Figure 4.4. A Standard Training Simulator Configuration

This configuration permits hands-on training in real-time plant operation and provides for the student to gain a feel of the process. Furthermore, it permits the development and implementation of both basic principle and full scope training simulators.

The extent of training scenarios possible depends on the relative detail embedded in the simulation model. If all manual operations, error models, etc. are incorporated in the simulation model along with the dynamic models of major plant components then it is possible to perform interactive plant start-up and shut-down, and also to train in operating the plant under normal and abnormal operating conditions.

With this configuration the user interacts with the simulation through the standard control system interface. However, the effective use of this configuration demands the presence of an expert instructor who will guide the student through a training programme. Additionally, the instructor has to provide all the necessary support and feedback. Furthermore, a trainer will typically set up configurations, inject disturbances, instigate other scenarios, etc. in a manual way, as required by the training objectives. The trainer interaction with the simulation system is performed from a separate trainer console.

4.2.2. A Simulation Based Training System

At a higher level of complexity the operations management module can be linked to the above configuration and lead to a simulation based system (Kassianides and Macchietto, 1990) which achieves quite advanced functionalities and offers more possibilities than the standard training simulator described above. The way these modules are interconnected is presented in figure 4.5. The simulation, control system, and operations management modules form the backbone of the system, with the control system module being the main one.

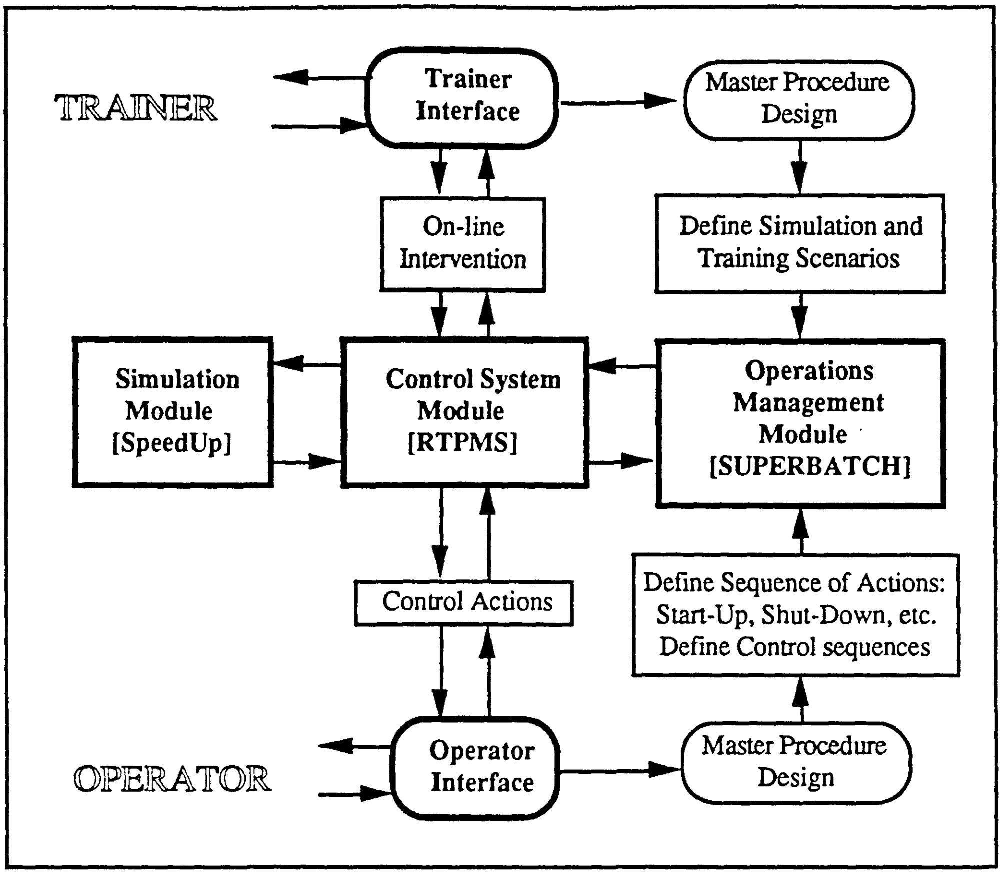


Figure 4.5. The Structure of the Simulation Based Training System

The control system also provides the user interface through appropriate consoles for on-line operator and trainer interaction. Alternatively, both the trainer and the user can predefine and save off-line operating procedures and sequences (for example, a sequence that will lead to plant start-up) in the way described in section 4.1.3. From the trainer console the trainer can initiate these predefined procedures and training sequences and schedule them to occur at pre-specified times. Alternatively, he can interactively schedule an event or make status changes to the system. The operator can also be assigned the capability of assembling, on-line, predefined procedures to produce complex schedules as required by the training session.

In the prototype pilot plant application (described in later chapters) the main features of the plant have been incorporated in a dynamic simulation model. The control interfaces used for training are an exact replica of those used to control the actual plant and all actions available to an operator on the actual plant have been described as control primitives in the operations management module. This extensive and detailed representation of the domain allows the simulation based training system to be used in a variety of ways.

First, the trainer's view is taken. The system has been used as a demonstration tool to show normal plant operation. In this *demonstration mode* the functionalities of the operations management module are used to "play out", in real time, the predefined training sequences which demonstrate correct operation, the effects of disturbances, error recovery procedures, etc. For the purpose of demonstration proper start-up and shut-down sequences have also been defined which when executed automatically indicate to the trainees effective ways to start-up or shut-down the plant. These demonstrations can be repeated as necessary until the operators feel comfortable with the processes and actions involved.

In an *interactive mode* the trainer is able to either introduce a disturbance on-line, or schedule such a disturbance (in the form of prespecified training scenarios) to occur at some prespecified time. A number of scenarios including fault sequences have been developed of varying degree of difficulty. Once a disturbance has been introduced the trainees are expected to notice a deviation from normal plant operation, identify the nature and source of the problem and take the necessary corrective actions.

The students can also use the interactive mode in different ways. First, they can formally define their own start-up and/or shut-down procedures and then initiate them and observe and critically evaluate their choice. Second, they can interactively choose one of the prespecified training scenarios and practice normal plant operation, fault diagnosis and fault handling.

The simulation based training system offers two main advantages over the standard training simulator configuration. First, the system can be used stand-alone. Actions, training scenarios, and training sessions can be easily described and stored by the trainer without great effort and at any time. Then the student can access these sessions and complete the training process on his own.

Second, the system allows for the individualization of the training process whether it is being used as stand-alone or with the support of an instructor. Each operator can use the system on his own and practice repeatedly a training sequence. He can develop and test operational sequences until he successfully meets any operational objectives for a specific training session. Furthermore, he can follow demonstrations and practice the interactive operation of the plant until he achieves a desired level of competence.

4.2.3. An ICAI Tutorial

The intelligent module described in section 4.1.4. can be used independently from the simulation based system as a platform for the development and delivery of CAI or ICAI. In the pilot plant application an ICAI tutorial was implemented which aims to provide the student with a qualitative understanding of the process and the plant through the presentation of theoretical material and through the presentation of steady state cause and effect relationships that define the plant operation. Its objective is to re-enact the interaction with expert instructors that takes place prior to the students being exposed to the simulation based training system or the actual plant.

The general structure of the *Tutorial* as implemented is shown in figure 4.6. The *Tutorial* is comprised of three training sessions: (a) an introductory, (b) an intermediate, and (c) an advanced training session. The objective of the introductory session is to introduce a novice student to fundamental principles. It introduces the trainee to the process and the various plant components and their characteristics. Then, the intermediate session builds on this knowledge to introduce the concept of plant subsystems and the functional and cause and effect relationships of components interacting within each subsystem. Finally, the advanced training session examines subsystem interactions until ultimately the whole plant is examined.

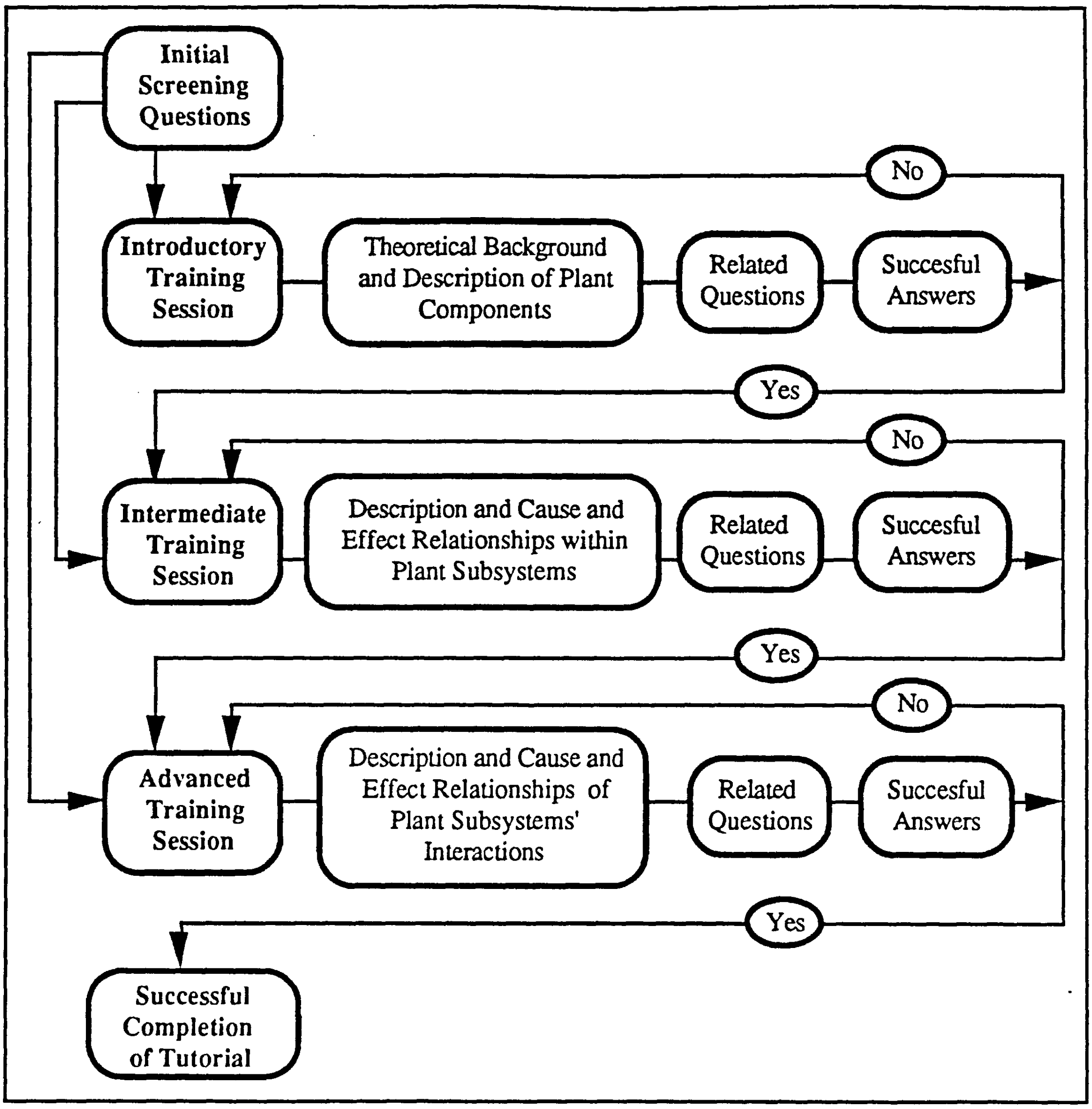


Figure 4.6. The General Structure of the *Tutorial*

The instructional material is presented in the form of both signal flow and simple mimic diagrams in conjunction with verbal descriptions. Furthermore, steady state simulations of individual plant components are available where thought to be appropriate. Steady state simulations of each of the plant subsystems is also provided, as well as steady state simulations describing the subsystem interactions.

Two different instructional strategies have been implemented in the *Tutorial*. First, a constrained instructional strategy has been implemented for process operator training. This forces the trainee to follow a predefined sequence through the training session material. The rationale behind this approach is the 'guarantee' that the student will be presented with all the necessary information and that important sections will not be overlooked.

Second, an exploration instructional strategy has been implemented for engineer training. While a suggested training sequence is presented to the student, he is allowed the flexibility to select what to study within each training session. In this way he/she can focus on particular problems and skip other subjects.

The instructional material for the two strategies is essentially the same, but the "instructional control" mechanisms differ. Whatever instructional strategy is followed the student has to successfully fulfil the requirements set by the respective student model prior to being allowed to proceed to a more advanced session.

4.2.4. An Integrated Computer Based Training System

An integrated computer based training system based on the proposed system architecture has been achieved (see fig.4.7.).

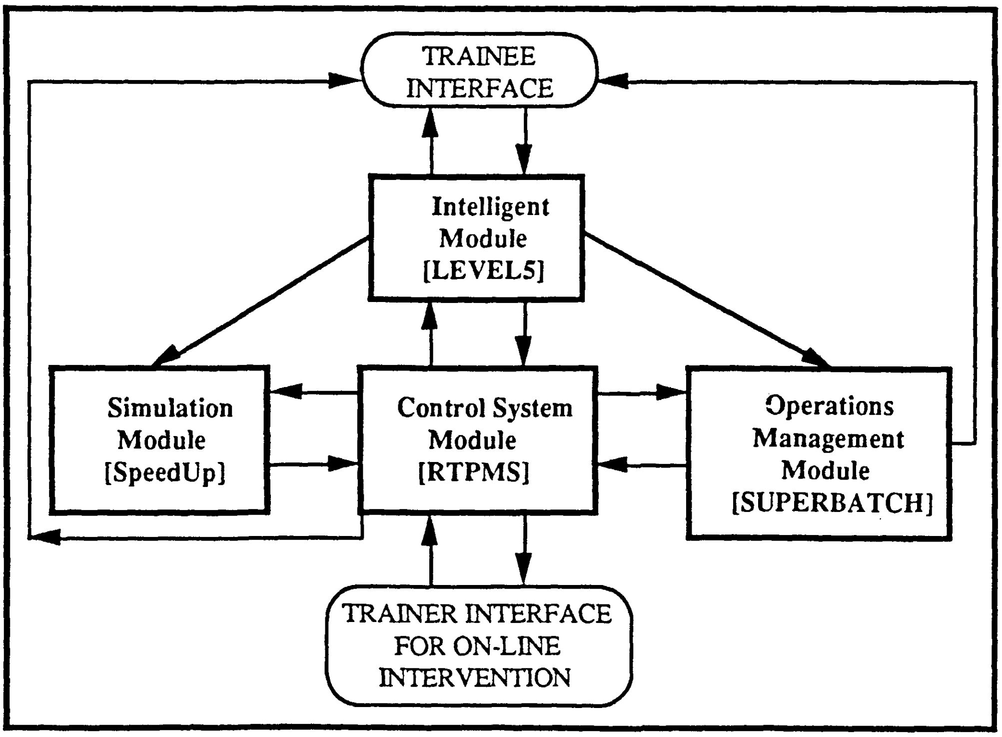


Figure 4.7. An Integrated CBT System for Operations Staff

This brings together all the features and advantages offered by both the simulation based training system and the tutorial system. The functional description of the communication between the various modules has been described in previous sections.

The main advantage offered by this system is its ability to combine qualitative information with 'hands-on' training in a realistic environment. The instruction is performed through the provision of qualitative explanations, guidance and advice while steady state and real-time simulations provide the means for acquiring 'hands-on' experience of plant operation. Furthermore, the instruction is delivered automatically and provides for the complete individualization of the training process in a stand alone system. However, the capability for trainer on-line intervention is retained at the control system module level.

For example in the case of the *Tutorial* the incorporation of real-time simulation as another instructional tool enables trainees to acquire a feel of the process dynamics at the component, subsystem and subsystem interaction levels. This re-enforces the mental model developed through the qualitative instruction by providing a direct experience of plant operation.

At the other end, the parallel presentation of qualitative information during the operation of the simulation based training system leads to more effective training since explanation of the dynamic effects and responses observed during normal or exceptional plant operation can be presented.

Finally, the integrated system enables the development of stand alone training programmes which take advantage of all the functionalities offered by each of the modules while at the same time minimizes the user development effort. Such programmes have been developed and are examined in the following chapters.

4.3. Hardware and Software Configuration.

The overall software and hardware of our set-up is given in figure 4.8. The simulation based training system has been developed and implemented on an IBM 4341 mainframe running a VM/CMS operating system. The version of RTPMS that has been used operates under the OS/VS1 operating system with special real time extensions, so one CMS virtual machine is configured for this purpose. However, SpeedUp and SUPERBATCH run in the VM environment.

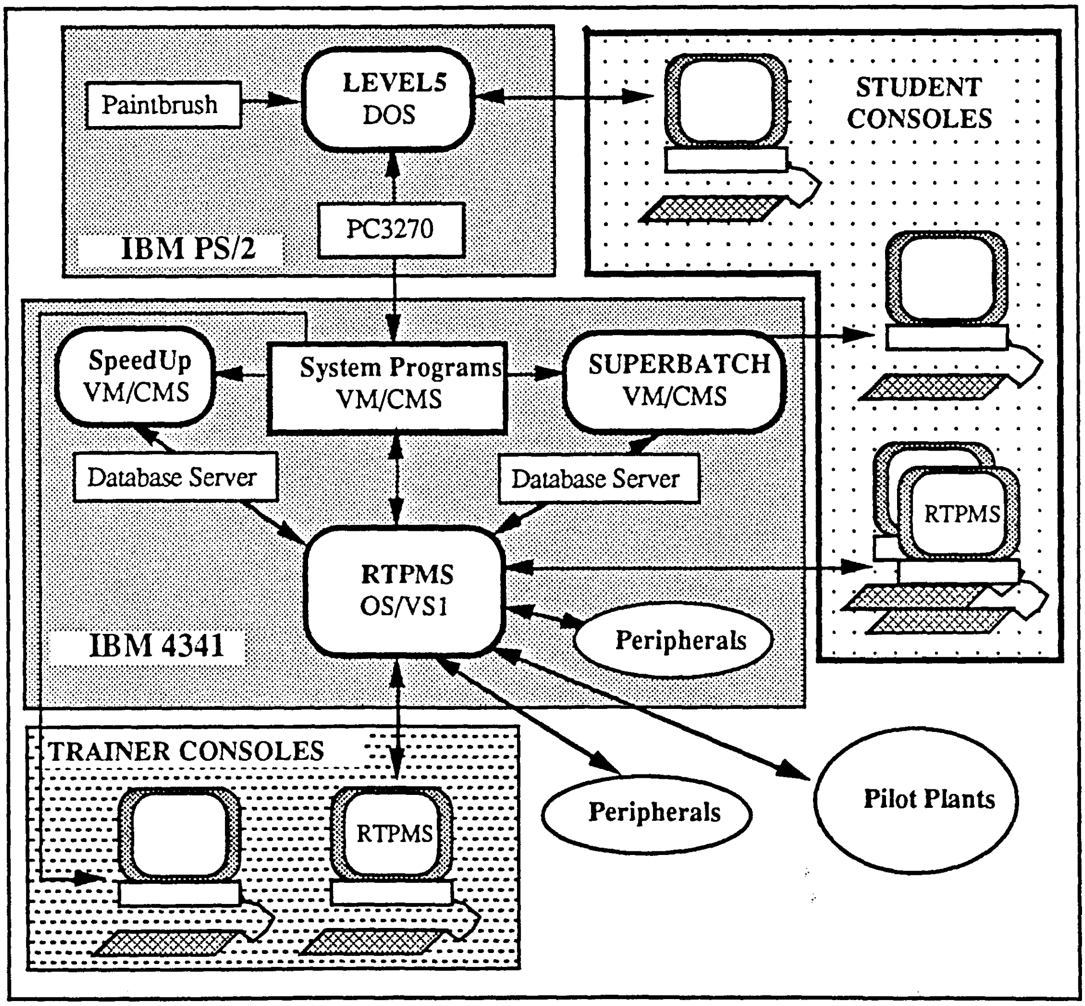


Figure 4.8. Software and Hardware Configuration

The communication between VM/CMS and OS/VS1 is achieved through special data base server (DBS) software (Macchietto et al. 1987). DBS supports a high rate of data transactions and makes it possible and very easy to link the real time RTPMS database with other databases, application programs, etc. in VM.

The intelligent module was developed and implemented using a personal computer version of LEVEL5 on an IBM PS/2 55SX under the DOS operating system. The PS/2 is also equipped with a PC 3270 emulation card and the related emulation software that enable the communication link between the DOS and the VM operating systems.

Currently the communication between the intelligent module and and the other modules is done through an intermediate CMS machine via file transfer. This hosts system programs that identify flags included in the commands issued from LEVEL5 and can then establish the proper information routes between the intelligent module and the other modules. Finally, Microsoft Paintbrush was used for the creation of the graphics displays that are used by the intelligent module.

4.4. Summary.

In this chapter an integrated system architecture was described which allows the development of flexible and cost-effective computer based training systems through the utilization of existing general purpose building blocks for modelling, simulation, control, operations, management, and expert system development.

The fact that each of the underlying tasks involved in the development of computer based training systems can be carried out in a software environment most suited to it decreases the development effort and thus could increase the cost-effectiveness of the integrated system architecture. While specific software components were used to implement and test this architecture it has been shown that the techniques are general and widely applicable. In view of this several functional requirements were indicated that could serve as a reference for the selection of the appropriate components.

Furthermore, the flexibility of the system was presented through a description of a number of training configurations that have been achieved with the proposed architecture. It has been possible to develop sophisticated simulation based training systems, structured tutorials, and systems which combine the features of CAI, ICAI, and training simulation. Furthermore, the training systems are easy to develop, flexible, easy to modify and easy to maintain. Additionally, the modular approach followed in the definition of the 'simulation problems' and the 'procedures' libraries allows the easy development of a range of training programmes.

Finally, the broad functionalities presented in section 3.7. are largely met by the capabilities of the integrated system that has been implemented. The need for the pre-training stages discussed in chapter 3 are still there but the tools to implement a training programme are available. In the following chapter three training programmes are described that use the system architecture for the development and delivery of training to process operations staff.

Chapter 5

PROCESS TRAINER: THE TRAINING PROGRAMMES

In chapter 4 an integrated system architecture was presented (from here on referred to as ISA), which it was claimed, permits the development and delivery of a variety of CBT courseware. This hypothesis is examined in the rest of the thesis. In particular, based on the general features of the ISA, three training programmes have been formulated which aim to provide the trainee with the background and experiences required for effective plant operation.

In this chapter these training programmes, each addressing a different training area, are defined and the general methodologies utilized are presented. The broad objective of the training programmes is to provide training to operations staff (engineers and operators) on several aspects involved in plant operation. In particular, three main areas have been addressed: (i) background training for plant operation, (ii) training in plant-operation, and (iii) training for fault identification.

Accordingly these have been incorporated as generic modules in *Process Trainer: A Tutoring System for Operations Staff,* a prototype 'expert system shell' for process training applications (*Process Trainer's* functional definition is given in chapter 6). They employ a variety of training methodologies, are stand-alone while also permitting on-line trainer intervention, and deliver instruction through a combination of theory and practice. Furthermore, the formal methodological definition of *Process Trainer* permits the easy definition and integration of additional training programmes.

An examination of the role of the process control operator (see chapter 2) indicated that in order to execute his tasks, he needs to draw information from different types of knowledge and perform a range of cognitive activities. The operator of a complex plant needs to recognise states and produce relevant responses as well as make inferences about the current and future states of the plant. Consequently, training is needed to provide and maintain these sorts of knowledge and to practice these types of activity. The types of knowledge involved and the range of cognitive activities performed are summarised in tables 5.1. and 5.2. (reproduced from Bainbridge (1990)).

Table 5.1. Types of knowledge that an operator refers to.

1. Product targets and plant constraints.
2. Interface -> process state inferences.
3. Process state -> response required.
4. Cause-effect relations.
5. Dynamic 'mental models' of how a given part of the plant works.
6. Actions available and their effects.
7. Typical sequences of events in the process, during:
    - start-up/shut-down
    - batch processing
    - after a given fault.

Table 5.2. The main cognitive activities of a process control operator.

1. Review/predict product demands.
2. Infer present state of the process.
3. Predict events, and associated states and actions.
4. Evaluate present or predicted state.
5. Review actions available and their effects.
6. Choose best action given multiple criteria.
7. Plan future actions.

5.1. The Part-Whole Training Method

Bainbridge (1990) outlining a general approach to operator training indicates that "as in training for any complex task, it is best to use the 'part-whole' method, teaching about the plant and task in a hierarchical way, starting with simple units and building up to how to understand and operate the whole plant". This method has been adopted for the formulation of the training programmes.

However, a range of training delivery methods has been employed throughout the programmes in order to meet the specific training objectives which are associated with each of the three main areas addressed. For example different approaches were used for the training on simple tasks and problem solving tasks.

The 'part-whole' method builds up the understanding of a complex task from the understanding of its components. Three ways of dividing a process into smaller sub-parts which are easier to understand are commonly employed (Bainbridge, 1990). These are:

1. Grouping parts of the plant according to whether they serve the same function.
2. Making divisions based on the task rather than the plant.
3. Dividing the process into unit operations, sub-parts of the plant between which the flows are relatively simple.

Breaking up the process according to the important functions that need to be maintained emphasizes their purpose to the student and focuses the instruction on their specific (possibly plant-wide) objectives. Understanding the process in terms of maintaining important functions is particularly relevant in plants which are sufficiently complex to have main functions which can be maintained by several methods. For example, several methods could be employed to maintain the cooling of a reactor system, to maintain a drum level, to maintain the supply of heat in a boiler, etc. This approach is particularly useful for training in fault handling and fault management where trainees "need to learn how to deal with unusual fault situations, as they need to maintain the main plant safety functions if the usual methods are not available due to a fault" (Bainbridge, 1990).

The second method for dividing the process into smaller parts that are easier to understand is based on the principles of task analysis. The importance and characteristics of task analysis as a pre-training step that helps to define the training objectives and focus the instructional process were indicated in section 3.4. Making the divisions of a process to be learned, based on the tasks to be performed rather than the plant, enables the development of a training programme that addresses both the operational objectives and the sequence of particular actions that an operator will have to make in order to achieve them.

Finally, dividing the process into unit operations and plant subsystems between which the flows are relatively simple, allows for the instruction to focus on the properties, characteristics, and, cause and effect relationships of individual blocks independently. Once these individual units are understood they can then be treated as building blocks and be assembled into larger independent units and so on.

Each of the three approaches of dividing the process has its advantages. However, in view of defining a training programme that can be successfully implemented from a generic CBT system it is necessary to consider the flexibility and generality of the methods. The first two methods discussed above are entirely plant specific or 'environment' specific as a function or a task is defined within the context of a specific plant. Therefore, even in the case of a general training facility, the knowledge and skills conveyed to a student about how to maintain a particular function or perform a specific task will be defined from the specific operational environment. These methods are therefore less effective in describing first principles or issues of general applicability.

On the other hand breaking up the process in units allows the general description of individual components that can then be assembled as necessary by the trainer in order to meet specific training requirements (these could extend to maintaining certain functions or performing specific tasks).

It is often possible to define as a major task the need to maintain an important function through the appropriate handling of unit operations. Therefore, an understanding of the various units is necessary in order to both maintain a particular function (such as maintaining cooling) or achieve a high level task (such as setting up the liquid circulation in a plant). Consequently dividing the process into units provides a general base for developing a range of CBT systems and so it provides the foundation for the *Process Trainer.*

5.2. Conveying the Background for Plant Operation

It was pointed out above that it is necessary for an operator to have an understanding of the properties and characteristics of the components that make up a plant, and the important plant-wide cause and effect relationships that determine a plant's operational behaviour. It is therefore thought that dividing a process according to plant units and plant subsystems is the most appropriate approach towards the formulation of a CBT system aimed at providing the necessary foundations for plant operation.

With this method a specific plant is divided into smaller 'units' of different levels of complexity. Then starting with the simplest units and going on to the more complicated ones, the instruction aims to convey each individual ones characteristics and functionalities. This is achieved by examining every unit in isolation, independently from the rest.

For example, first low level units (such as valves, controllers, etc.) are taught, then medium level units (such as heat exchangers, feed tanks, etc.) and last high level units, which are formed from the combination of low and medium level units into larger interdependent combinations of plant subsystems (such as heat exchanger subsystems, feed subsystems, etc.). Finally, the interaction between interdependent plant subsystems (for example the interaction between the feed and the heat exchanger subsystems) is examined. Representative 'units' of a plant are listed in table 5.3.

Table 5.3. Representative plant units.

1. **Lowest level:** valves, pumps, transducers, controllers, etc.
2. **Medium level:** tanks, condensers, heat exchangers, splitters, etc.
3. **Highest level:** combinations of unit operations which are interdependent in plant subsystems: heat exchanger subsystems, feed subsystems, etc.

Once the process has been divided into smaller units the instructional material for each one needs to be organised. It is appropriate (Bainbridge, 1990) to first convey the general cause and effect relations and then to teach a feel for the dynamic characteristics of individual units such as gains, lags, etc. Therefore, the sophistication needed of the training facilities is different at each level. The general cause and effect relations could be introduced by means of textual information, dialogues, diagrams, steady state simulations and perhaps dynamic simulations. The capability of the simulation based system to speed-up or slow down the execution of a dynamic simulation problem can be used here to show the overall dynamic characteristics of a process unit.

In contrast, training aimed at providing a 'feel' for the dynamics of the process unit could only be provided by a high-fidelity simulator (see chapter 3). Therefore, it is important that the simulation based system is capable of providing realistic representations of the process dynamics in real time. In this situation the simulator is the main instructional tool even if the training is supplemented with help, guidance and advice facilities provided as textual information, diagrams, etc.

Bainbridge (1990) presents three possible principles for sequencing the material which the trainees experience:

1. Building up from simple to multiple unit plant operations.
2. Building up understanding starting from simple plant dynamics and moving onto more complex ones:
    - Starting with operations with short time constants, moving on to ones with long time constants.
    - Starting with operations with a small number of variables, moving to ones with a larger number of variables.
    - Introducing complexity of dynamics in the sequence:
        - effects of manipulated variables which are additive in effect, i.e. the level of any one does not affect the effect of another.
        - effects of manipulated variables which are interactive, i.e. the level of one does affect the effect of another.
        - separate unit operations which are interdependent.
3. Teaching functions in order of importance, starting with main function, going on to subsidiary functions, and ending with maintenance of services.

These broad specifications for a training programme aimed at conveying the background necessary for plant operation, led to the formulation of the *Tutorial.*

The Tutorial

In the *Tutorial* the trainee is first introduced to the various plant components and their characteristics. Then, the *Tutorial* builds on this knowledge to introduce the concept of plant subsystems and their respective dynamics and cause and effect relationships. Finally. subsystem interactions are examined until ultimately the whole plant is examined.

The *Tutorial* is thus divided in three training sessions: (i) an introductory or Plant Components session, (ii) an intermediate or Plant Subsystems session, and (iii) an advanced or Plant Subsystem Interactions session. The general structure of the *Tutorial* was given in figure 4.6. Its functional definition is described in chapter 6 while an example application is presented in chapter 7.

The Introductory Training Session

The introductory training session is geared towards novice student operators. It assumes that the student is not familiar with the process, the plant, or the plant components. Therefore, the training objective of this session is to convey to the student an understanding of the process and the plant components that will enable him to draw some inferences about the inter-relationships that make up the plant. The subjects covered by the introductory training session are listed in table 5.4.

Table 5.4. Introductory Training Session Subjects

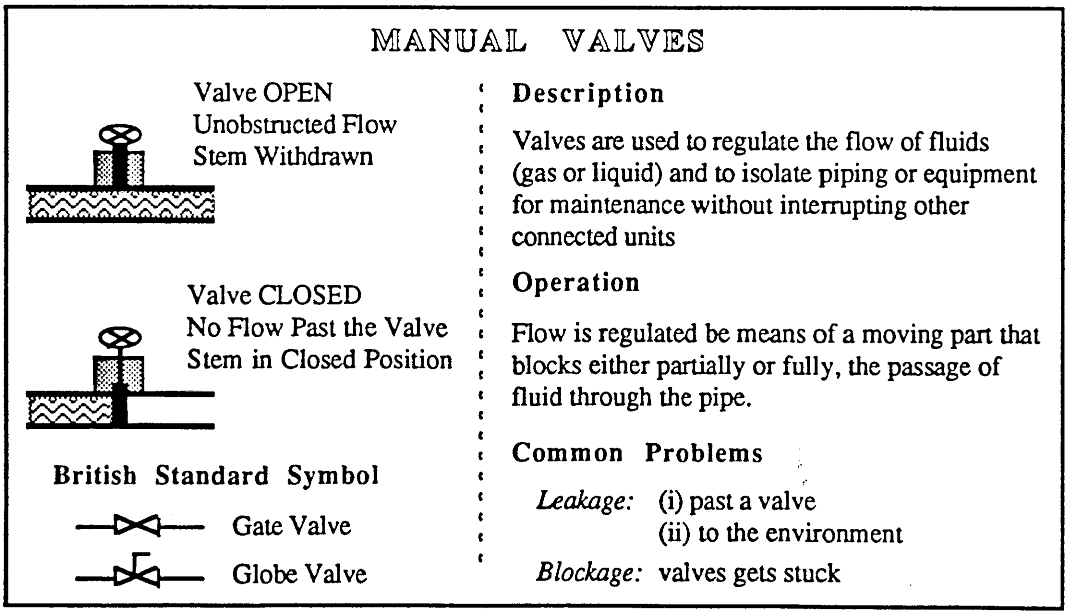
1. **The Process**
2. **The Plant**
3. **The Plant Components**
    - Low level 'units'
    - Medium level 'units'

Initially in the introductory session the student is given a description of the process. This is achieved through the presentation of appropriate diagrams which are accompanied by explanatory text. In the second step of the instructional process the student is introduced to the general plant structure through the presentation of a diagrammatical flowsheet of the complete plant layout. At this point the student is informed that by the end of the session he should be able to identify the various plant components, be familiar with their individual properties and characteristics, and be able to follow the process path through the plant components. In this way, the end objective is presented to the student.

The next step in the introductory session is the examination of each individual plant component (low level and medium level units). The descriptions provided are as simple as possible and serve only to provide the student with a general understanding of the fundamental characteristics of individual units as they may relate to plant operation. Nevertheless a variety of presentation methods and training facilities are used depending on the level of the unit described.

Low level units are treated in an entirely qualitative manner through the use of diagrams and textual information. Their examination consists of: (i) a description of the functional properties of the unit (what is it?), (ii) a description of the operational characteristics of the unit (how does it work?), (iii) a presentation of common problems associated with the unit (what can go wrong with it?), and (iv) a presentation of the British Standard Symbol used to represent the unit in a flowsheet.

Verbal descriptions are used for both the description of the functionalities and the uses of the unit, and the common problems that may be associated with it. However, the description of the operational characteristics is done through a combination of a verbal description and simple schematic diagrams. An example display describing a manual valve is shown in figure 5.1.



MANUAL VALVES

Valve OPEN
Unobstructed Flow
Stem Withdrawn

Valve CLOSED
No Flow Past the Valve
Stem in Closed Position

British Standard Symbol

Gate Valve
Globe Valve

Description

Valves are used to regulate the flow of fluids (gas or liquid) and to isolate piping or equipment for maintenance without interrupting other connected units

Operation

Flow is regulated be means of a moving part that blocks either partially or fully, the passage of fluid through the pipe.

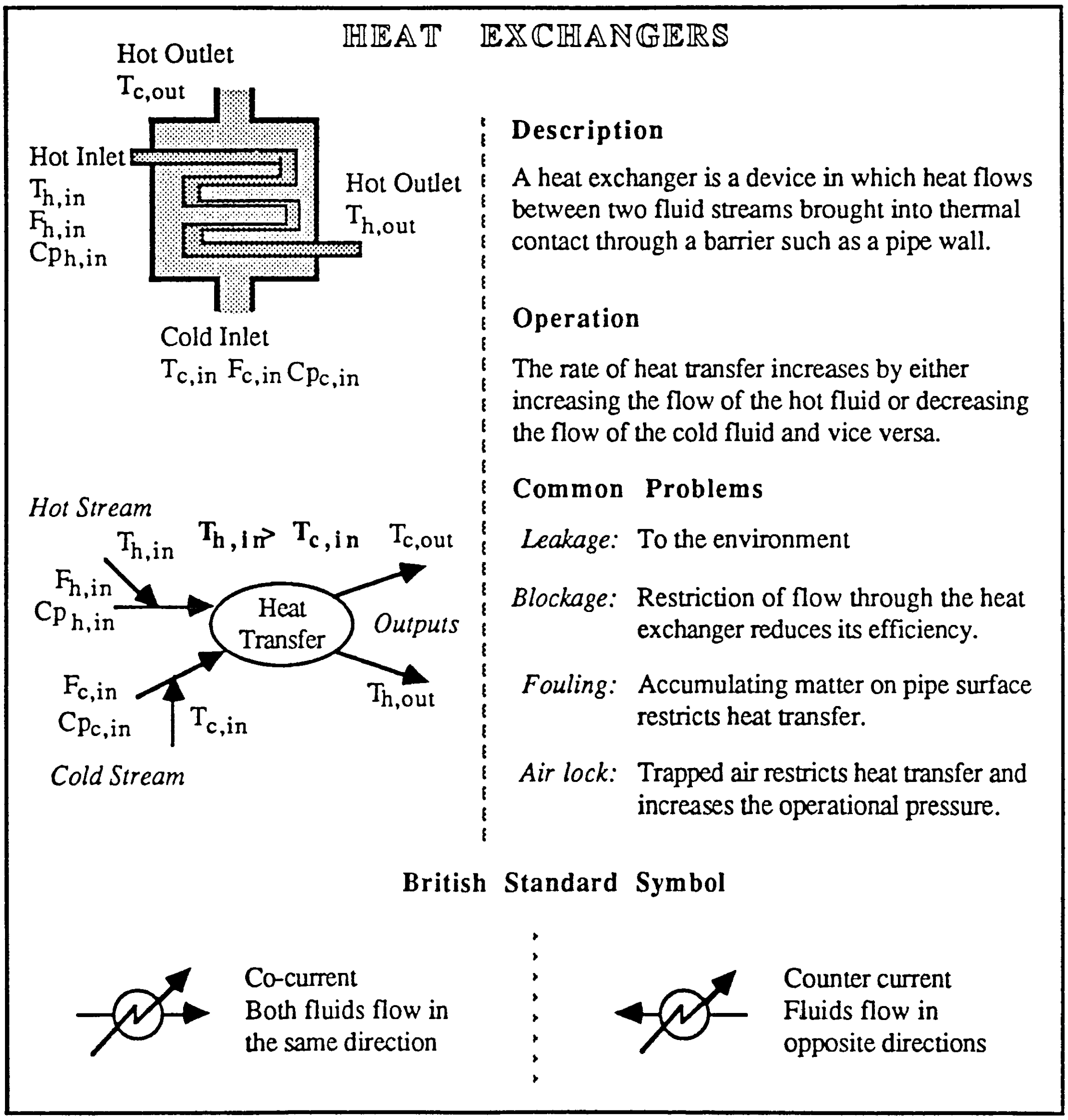
Common Problems

Leakage: (i) past a valve
(ii) to the environment
Blockage: valves gets stuck

Figure 5.1. An example display describing a manual valve

The same four factors are also used for the examination of the medium level units. However, contrary to the low level units, medium level units are treated both qualitatively and quantitatively.

The format used for the qualitative examination of these units is an augmented version of the one used for low level units (an example display describing a heat exchanger are shown in figure 5.2.). As with the low level units verbal descriptions are used for describing both the functional properties of each unit and the common problems that may affect its normal operation.



HEAT EXCHANGERS

Description

A heat exchanger is a device in which heat flows between two fluid streams brought into thermal contact through a barrier such as a pipe wall.

Operation

The rate of heat transfer increases by either increasing the flow of the hot fluid or decreasing the flow of the cold fluid and vice versa.

Common Problems

Leakage:	To the environment
Blockage:	Restriction of flow through the heat exchanger reduces its efficiency.
Fouling:	Accumulating matter on pipe surface restricts heat transfer.
Air lock:	Trapped air restricts heat transfer and increases the operational pressure.

British Standard Symbol

Co-current
Both fluids flow in
the same direction

Counter current
Fluids flow in
opposite directions

Figure 5.2. An example display describing a heat exchanger

However, the operational characteristics of medium level units are presented through a combination of mimic diagrams, cause and effect graphs and verbal descriptions. Mimic diagrams are best in showing the topology of the physical structure and as such map directly on the plant flowsheet that the student will use when actually operating the plant. On the other hand, cause-effect diagrams focus on making the causal structure explicit.

It has been suggested (Bainbridge, 1990) that "when someone does not know the cause-effect relations in the plant, then he can best solve problems about how the plant works by referring to a cause-effect network. However, if he does know how the plant works, then a mimic provides richer information. The symbols, and relations between parts of the process, provide cued reminders to knowledge about how the process works, which are not explicitly mentioned in the cause-effect diagram." Therefore, the use of both types of diagrams make the descriptions of the units suitable for students having different levels of experience.

The quantitative examination of the medium level units is done through the use of appropriate steady-state simulations. The student can study the steady-state properties of a unit by manipulating its inputs and observing the corresponding outputs under steady-state operation. Alternatively, the student may choose to change certain design specifications of the unit in order to examine how these affect its overall performance. As a rule the first time a student uses the simulation facility in order to study a particular unit he is guided through a few representative examples which indicate the basic principles for that problem. However, once those are completed the student is allowed to experiment on his own, noting the effects of his own actions on the unit's steady-state performance.

Figure 5.3. shows a sample interface to a steady state simulation of a heat exchanger. Here the student may change any or all variables associated with the two input streams (flowrates, temperatures, and specific heat capacities) as well as the minimum temperature difference which determines the maximum amount of heat exchange. Once the student has made his changes the execution of the steady-state

simulation problem returns new values for the outputs. In this way the student can experiment in a flexible way and examine a range of operational situations.

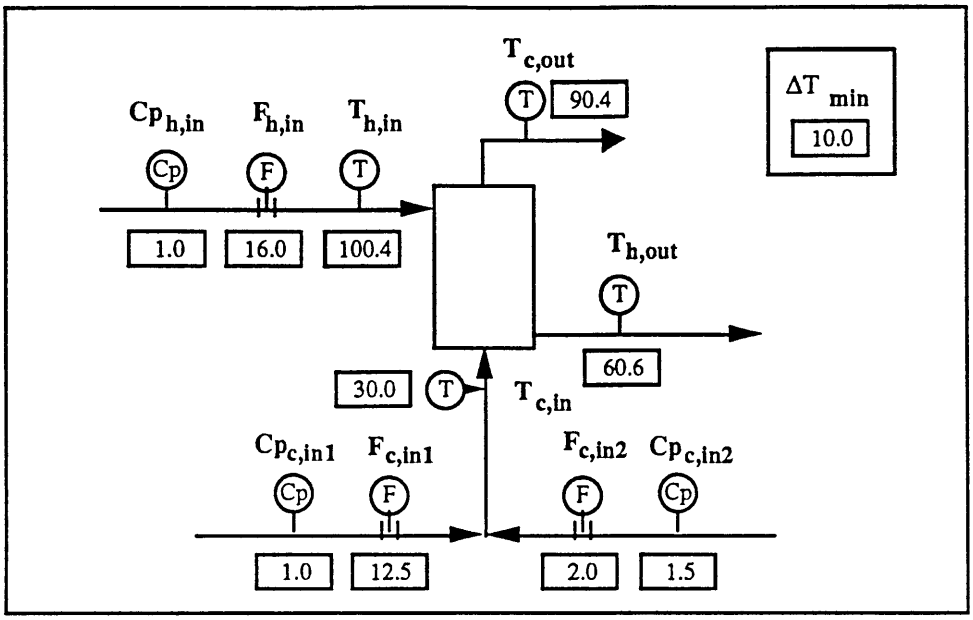


Figure 5.3. Interface to a heat exchanger steady-state simulation

The sequence in which the instructional material is presented to the student depends on whether he is undergoing training as an operator or an engineer. The difference in the two approaches is a result of the extent of their respective responsibilities.

Generally, a plant operator is expected to be able to monitor the status of the process and execute well defined procedures. Therefore, the training need is usually for an operator to learn some standard behaviour or cover a specific set of subjects in minimum time. This can be achieved if the sequence of the instructional material that the trainee experiences has been determined by the trainer at some time prior to the instruction.

This sequence can be defined according to a 'need to know' basis and the difficulty of individual topics (in this case units). Furthermore, the enforcement of a particular sequence should guarantee that all subjects have been fully covered by the operator, thus minimizing any possibilities of instructional material being skipped. In this way each operator is presented with exactly the same material in a formally defined and structured way.

In contrast to the operator an engineer needs to learn to 'understand' the system, be able to deal with unforeseen situations, and devise the procedures that the operator will have to execute. Therefore, contrary to operator training, where a more rigid path is followed through the instructional material, engineer training is more flexible and allows (guided) exploration of the instructional material. While a degree of exploration is incorporated within the instructional material (for example, in the steady-state simulations) for both the operators and the engineers, the flexibility factor is greater for engineers since they can choose which subjects to study, which to skip, and the order in which they will study them.

Finally, there is the question of assessing the student's understanding of the subjects covered in the introductory session. This is achieved in two stages. The first stage comes immediately after the instructional material for a particular unit has been covered. At this time a set of questions relating to the unit's functionalities and characteristics is provided which the student has to answer correctly prior to being allowed to study another unit. In this stage feedback is used both to congratulate correct answers (positive reinforcement) and to give information to the trainee explaining wrong answers on the basis of which they can understand their mistake and improve their next try.

The second stage comes after all subjects covered in the introductory session have been studied. In this stage a set of questions covering material that has been studied in the entire session is presented. Then, based on the success of the students answers, areas of deficient performance are identified. Isolated wrong answers are treated in the same way as that described for the first stage. However, a set of wrong answers consistent to a particular unit leads to the student being directed to repeat the instructional material for that unit. Finally, before a student can continue with the intermediate session of the *Tutorial* he has to repeat, and show proficiency in the units which he had not sufficiently understood.

The Intermediate Training Session

The training objective of this session is to convey an understanding of the cause and effect relationships and the functional dependencies within each plant subsystem. The intermediate or Plant Subsystems training session builds on the understanding of the properties and characteristics of individual plant components by assembling these components into high level units which are then examined in isolation. It is assumed that the student about to begin this session has either successfully completed the introductory training session of the *Tutorial* or he has acquired 'knowledge' of the plant components in the specific plant in a different instructional setting (for example through a lecture).

A representative plant subsystem is shown in figure 5.4. This is the heat exchanger subsystem of a specific application described in more detail in chapter 7. As seen this subsystem is made up by a combination of low level units (manual valves, concentration meters, etc.) and a medium level unit (a heat exchanger unit).

Similarly to the examination of the medium level units the inter-relationships within a plant subsystem are demonstrated and explored through the use of both qualitative and quantitative methods. However, both the nature of the information conveyed and the format of presentation used in this session are different from the introductory session.

The qualitative examination of a plant subsystem consists of: (i) a description of the plant subsystem (which plant components make up the subsystem?), (ii) a description of the cause and effect relationships within the plant subsystem (what affects the operation of the subsystem?), and (iii) a description of common problems that may arise in the subsystems operation (what can go wrong?).

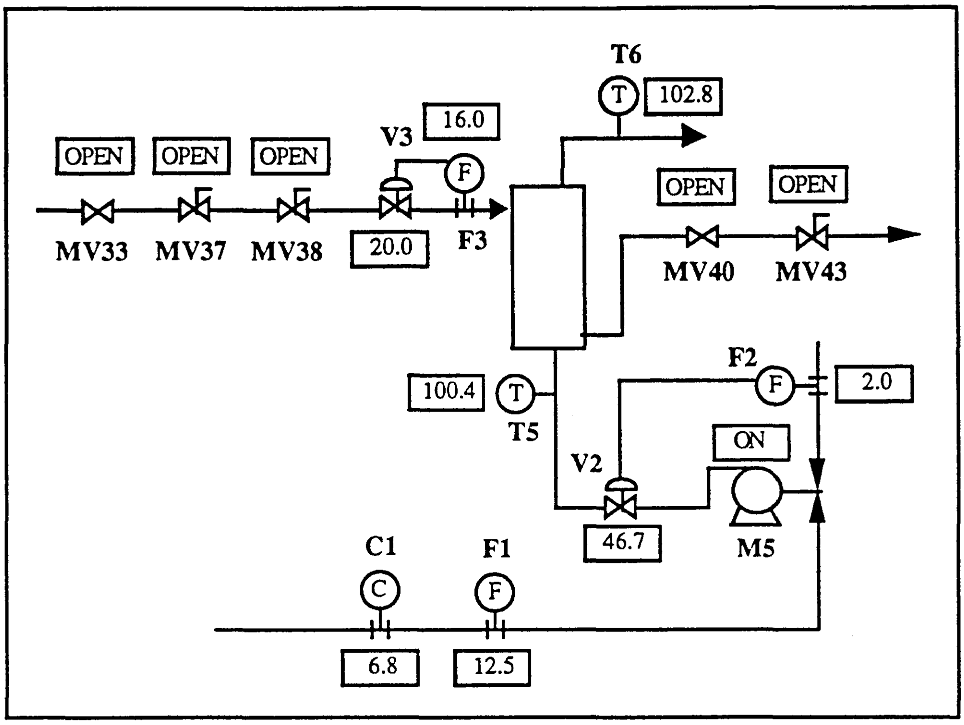
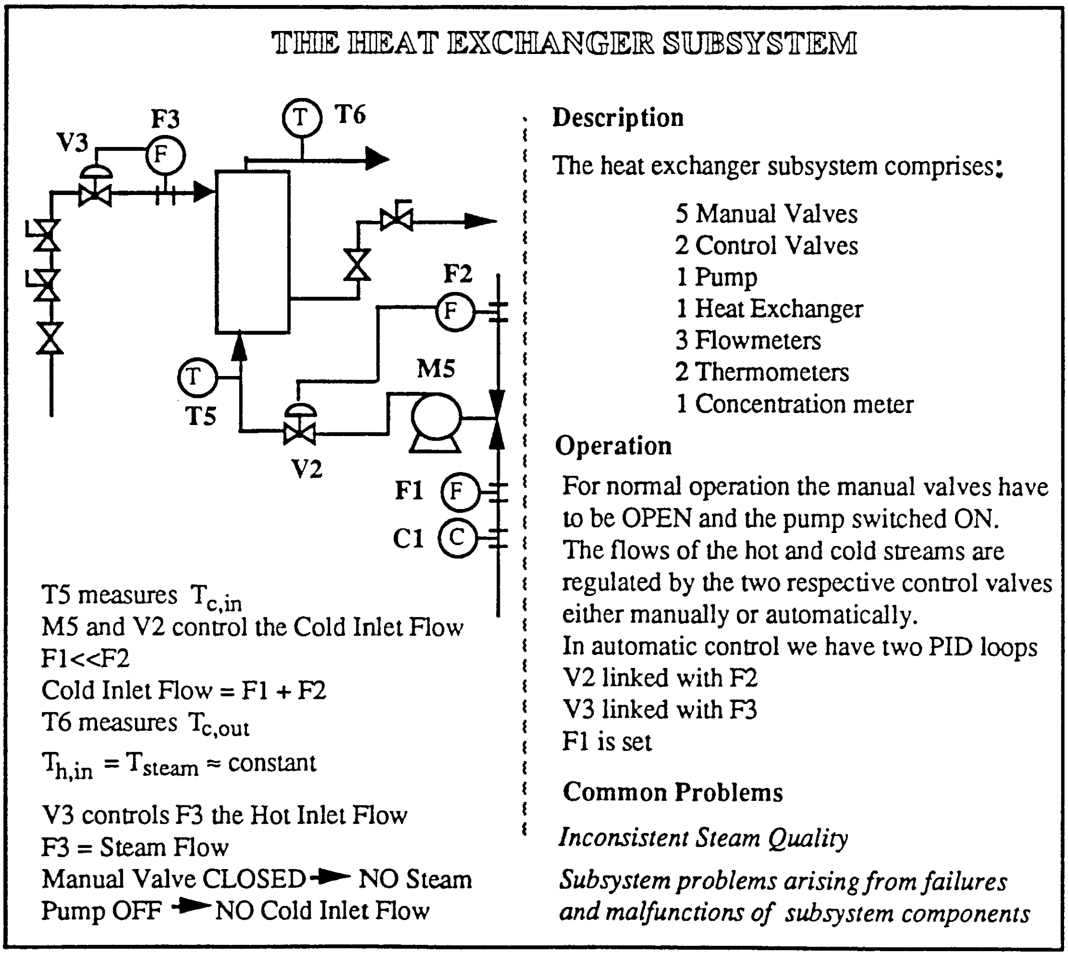


Figure 5.4. The heat exchanger subsystem of the Evaporator Pilot Plant (Interface to a steady-state simulation)

The description of the plant subsystem is provided verbally and through the use of appropriate mimic diagrams. The verbal description consists of a list of the plant components while the mimic diagram is typically an extract of the complete plant flowsheet showing in detail all plant components. This method serves two purposes. First, it provides the description of the subsystem and second it provides cues for the material studied in the introductory session.

The cause and effect relationships within the plant subsystem are also presented through the use of text and diagrams. However, in this case the verbal description of the functional dependencies is supported by appropriate cause and effect diagrams. Finally, the common problems that may occur within the plant subsystem are verbally described. Figure 5.5. shows an example display of the qualitative examination of the heat exchanger subsystem mentioned above.

The quantitative examination of plant subsystems is done through the use of appropriate steady-state and dynamic simulations. The use of the steady-state simulations is similar to that described for medium level units in the introductory session. Once again the student is guided through a few important representative examples before being allowed to explore on his own initiative the operational characteristics of the subsystem.



THE HEAT EXCHANGER SUBSYSTEM

T5 measures $T_{c,in}$
M5 and V2 control the Cold Inlet Flow
F1<<F2
Cold Inlet Flow = F1 + F2
T6 measures $T_{c,out}$
$T_{h,in} = T_{steam} \approx$ constant
V3 controls F3 the Hot Inlet Flow
F3 = Steam Flow
Manual Valve CLOSED → NO Steam
Pump OFF → NO Cold Inlet Flow

Description

The heat exchanger subsystem comprises:

5 Manual Valves
2 Control Valves
1 Pump
1 Heat Exchanger
3 Flowmeters
2 Thermometers
1 Concentration meter

Operation

For normal operation the manual valves have to be OPEN and the pump switched ON.
The flows of the hot and cold streams are regulated by the two respective control valves either manually or automatically.
In automatic control we have two PID loops
V2 linked with F2
V3 linked with F3
F1 is set

Common Problems

Inconsistent Steam Quality

Subsystem problems arising from failures and malfunctions of subsystem components

Figure 5.5. A display describing a heat exchanger subsystem

However, the simulation models in this session are more elaborate as they need to represent both the characteristics and the complexity of the real plant subsystem. Furthermore, since the emphasis is on conveying an understanding of a particular plant subsystem rather than first principles the student is only able to manipulate variables which affect plant operation and which may be accessed either directly or indirectly on the actual plant. Nevertheless variables that are not instrumented on the real plant and therefore not available on the real interface are shown if they are needed to enhance the understanding. A sample interface for the study of the steady-state operation of the heat exchanger subsystem was given in figure 5.4.

For example, in the heat exchanger subsystem the student may alter the status of the low level units (OPEN or CLOSE a manual valve, TURN a pump ON or OFF, etc.) and/or change the inputs to the medium level unit (the heat exchanger). Two distinct cases are identified for manipulating the inputs to the heat exchanger. The first involves changing the value of an input which is not controlled directly within the subsystem. In this case the student specifies directly the new value for the input. In the example representative variables are C1, F1, and T5.

The second case involves the manipulation of inputs that are directly controlled within the subsystem (in the example F2 and F3). This leads to two possibilities: manual control and automatic control. In the case of manual control a student can only regulate the value of an input by manually operating a control valve. Therefore, he may examine the effect of the valve position on the respective input. In contrast, under automatic control the student manipulates the input to the heat exchanger by setting the appropriate setpoint on the controlled variable.

The second type of quantitative examination of plant subsystems is through the use of dynamic simulation. The dynamic simulation models are consistent with the models used for the steady-state simulation. Furthermore, dynamic simulation is used in the same way as steady-state simulation. However, the emphasis here is placed on the actual dynamic behaviour of the plant subsystem with its response to input changes being representative of that which will be experienced in the actual plant.

Therefore, this training facility is used to convey to the student a 'feel' for the subsystem, by illustrating the effect of time constants, lags and gains, and also by examining in detail the characteristics of the control loops which are configured within it. For example, the dynamic simulation will exhibit what happens after a step-change is made to the setpoint of a controlled variable. It may indicate the oscillatory behaviour of a control valve, the time taken for the subsystem to reach the new steady-state position, etc.

The sequence in which the instructional material is presented to the student follows the same principles as those followed in the introductory session. Once again a difference is made between a rigid path for operator and a flexible path for engineer training. However, a degree of flexibility is also allowed for both categories of students at different points in the instructional process.

When studying a particular plant subsystem both the engineer and the operator student are initially presented with the qualitative description of its characteristics. Then the operator is directed to study the steady-state operation of the subsystem. However, before he begins he is given the opportunity to review the description of individual low and/or medium level units. When the steady-state operation of the subsystem has been studied he is directed to study the dynamic operation. Once again he is given the opportunity to review any material he thinks necessary before he begins.

In the case of the engineer a higher degree of flexibility is provided. Once the qualitative description of the subsystem has been studied an engineer is given the same number of alternatives, however, collectively. He may choose to: (i) study the steady-state operation of the subsystem, (ii) study the dynamic operation of the subsystem, (iii) review the qualitative description of the subsystem, or (iv) review the description of selected low and/or medium level units.

Finally, the evaluation of the student's understanding of the cause and effect relationships within individual plant subsystems follows a similar pattern to that described in the introductory session. The main difference here arises from the fact that understanding is tested at the subsystem level. Nevertheless, depending on his performance, a student may be directed to repeat the instructional process for even low or medium level units. This would then involve both the study of the relevant instructional material and the demonstration of adequate understanding of those unit(s) in the way described in the introductory session.

The Advanced Training Session

It is assumed that a student beginning this session has either successfully completed the intermediate session or that he has been conveyed the understanding of the individual plant subsystems in a different instructional setting. The objective of the advanced or Plant Subsystem Interaction training session is to convey to the trainee an understanding of the interactions between the individual subsystems of the specific plant. This is achieved by building on the knowledge that the trainee has gained from the two previous sessions.

It is possible to consider the combination of plant subsystems as a higher order plant subsystem comprised of high level units. The main difference is that some of the inputs to a subsystem downstream are the outputs of the subsystem upstream and therefore cannot be directly manipulated by the student. An example combination of two subsystems which is examined in the specific application described in chapter 7 is given in figure 5.6.

In this example we see that some of the inputs to the heat exchanger subsystem are outputs of the feed subsystem. In particular F1 and C1 are used as direct inputs while T1 has a direct effect on the input T5. Therefore, it is necessary to describe this new causal structure by combining the cause and effect relationships of the individual subsystems and accomodating the interdependences that define their interaction.

Another difference relates to the structure of the instructional process in the advanced session. First, the interaction of two linked subsystems (where the outputs of the one upstream are the inputs of the one downstream) is described. Then, following the same approach a third high level unit downstream is added to the first two subsystems, and so on, until the interactions in the whole plant are examined. However, the examination of the interactions between particular plant subsystems follows the same format and the same sequence of presentation as with individual plant subsystems. Once again, the qualitative description is provided by combinations of appropriate verbal descriptions, mimic diagrams and cause and effect diagrams.

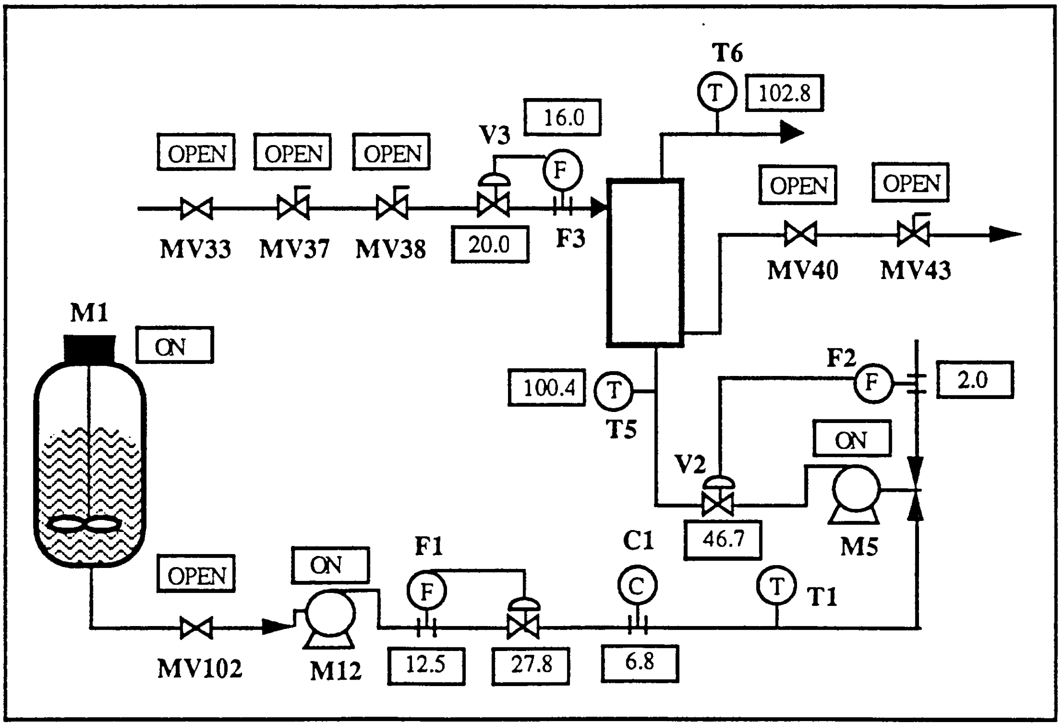


Figure 5.6. The combination of the feed and the heat exchanger subsystems (Interface for steady state simulation).

The quantitative examination of the subsystem interactions utilizes steady-state and dynamic simulations. The simulation models of the individual subsystems are combined and executed as one integrated model, which is used in exactly the same way as the simulations of individual plant subsystems in the intermediate session. Figure 5.6. shows the interface to the steady-state simulation exhibiting the interactions between the two subsystems described above.

The facility to review material offered to the student in the intermediate session is also available in this session, in exactly the same way and at the same instance in the instructional process. However, in the advanced session the student may also choose to repeat the instructional material of the individual subsystems involved in addition to the material relating to the low and medium level units. Furthermore, the same criteria apply for directing the student to repeat material after an evaluation session. A student may be directed to repeat instructional material relating to low or medium level units or he may be directed to review a section which describes one of the high level units.

5.3. Training for Plant Operation

The training programme described in the previous section aims to convey to operators the background necessary for operating a plant through the examination of the properties of the 'units' that make up the plant and the various cause and effect relationships that prevail in the plant structure. However, it does not examine what is involved in the actual operation of the plant.

A programme that aims to train operators in the actual operation of a plant is presented in this section. This examines the operational objectives and the sequence of particular actions that an operator will have to make in order to achieve them. For this purpose the division of the process into smaller sub-parts which are easier to understand is based on the tasks to be performed rather than on the plant units.

Since hands-on training is the main need addressed by this programme, the training is structured and based upon the availability of a training simulator facility. The merits, short-comings and characteristics of simulation based systems were examined in detail in section 3.5.5.6. Furthermore, in chapter 4 an ISA was described which provides the platform for the development of such systems of advanced functionalities. Here we describe the structure of a general training programme for plant operation. Its application to a specific case study using the facilities of the ISA is described in chapter 7.

Plant Operation Training Session

During plant operation an operator has to fulfil several operational criteria described as product quality/quantity, plant efficiency and plant integrity. These objectives are achieved through the choice and implementation of the appropriate control actions and through the implementation of sequences of actions. Consequently, an effective training programme for plant operation should address these needs.

Dividing the process based on the tasks to be performed permits the formal definition of operational sequences that will meet the operational objectives. Thereafter, an operator needs to learn the general form of the sequence and the actual order in which actions ought to be performed. He also needs to learn to identify the state that the process is in by interpreting the information that appears on the interface and to identify whether a transition state has been reached. Furthermore, an operator should be taught the reasons for the sequence.

Finally the operator needs to be taught how to do the sequencing. There are three aspects (Bainbridge, 1990) involved in the execution of a sequence: (i) carry out control actions to attain the state from which the next phase can be initiated, (ii) recognise when the state from which the next phase can be initiated has been achieved, and (iii) initiate the next phase if it is not initiated automatically.

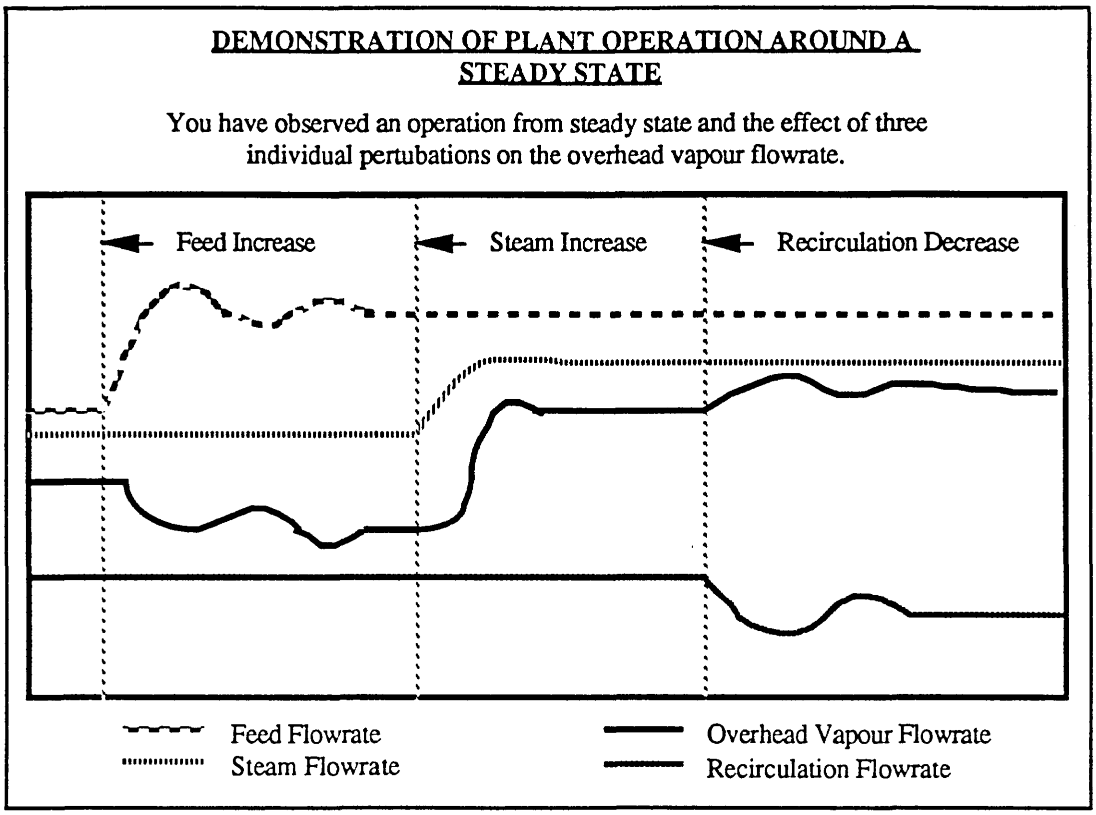
A special case arises in the training of engineers. While they need to learn what the operators are taught in view of operating the plant they also need to know how to devise the sequences of operations. Under normal operation an operator is expected to execute a predetermined sequence that will achieve certain operational objectives. This extends to abnormal operation.

While some degree of improvisation may be necessary in situations where an unforeseen event occurs, an operator is normally expected to react to an emergency situation by following predefined sequences of operations that will maintain plant integrity. These sequences may involve switching to an alternative (secondary) process path, executing an emergency unit or plant shut-down sequence, etc. In all these cases the sequences have been formulated by engineers and are based on knowledge and experience usually gained through the exploration of the properties of the process.

In order to meet the above requirements a combination of demonstrations and 'hands-on' experience training scenarios is used. In a demonstration mode an operational sequence is explained and then demonstrated to the student. This is achieved in several stages. First, the general objective of a task is explained and then the associated sequence of operations is presented in its general form. Following that the reasons for choosing this particular sequence are explained along with a presentation of a selection of other (second-best) sequences that could achieve the same purpose.

Following this qualitative examination of the sequence the operator is shown a demonstration of the sequence. This involves the real time implementation of the sequence in the realistic environment provided by the training simulator facility. In this demonstration mode the operator is shown how the sequencing is done and also what appears on the interface during the execution of the sequence.

The extent of possible demonstrations depends on the training objectives and on the previous experiences of the student with the plant. Therefore, building a set of experiences in the context of performing plant specific tasks may involve the demonstration of a whole sequence (such as for start-up, shut-down, emergency procedures, etc.) or part sequences which are similar to examining the cause and effect relationships in the whole plant. An example display (taken from the application described in chapter 7) which sums up the results of a demonstration of the effect of three individual perturbations on one specific variable is presented in figure 5.7.



DEMONSTRATION OF PLANT OPERATION AROUND A STEADY STATE

You have observed an operation from steady state and the effect of three individual pertubations on the overhead vapour flowrate.

Figure 5.7. Sample screen summarizing a real time demonstration

Once the operator is shown the demonstration of a sequence he should get hands-on experience in actually implementing it. This is also achieved by using the training simulator facility. In an interactive mode the student is given a set of operational objectives that in turn specify the tasks he has to perform and then he is given the sequence which he has to implement.

First the student is taught how to operate the computer based interface, for example how to get different displays, how to make control actions via the displays, etc. Then he has to use the interface to execute the specified schedule. Figure 5.9. shows an example display that outlines the operational objectives for a start-up of the evaporator pilot plant application described in chapter 7 and figure 5.8. outlines the resulting start-up sequence that will achieve the operational objectives.

INTERACTIVE PLANT START-UP

TURN THE POWER SUPPLY ON

1. Switch XL E 001 ON
2. Switch XL E 002 ON

TURN THE AIR DRIER ON

3. Switch XL E 003 ON

TURN THE AGITATOR ON

4. Switch XL M 001 ON

BEGIN THE COOLING WATER FLOW

5. OPEN XL MV 046
6. OPEN XL MV 047
7. OPEN XL MV 048
8. Close Cascade on XL V 004
9. Put setpoint of 1 kg/s on XL F 005

ESTABLISH THE LIQUID CIRCULATION

Start the Feed Flow

10. OPEN XL MV 102
11. Switch XL M 012 ON
12. OPEN XL V 001 (20%)

Begin the Recirculation

13. OPEN XL V 002 (40%)
14. Switch XL M 005 ON

Wait until XL L 001 begins to increase

Enable the Product Flow

15. Switch XL M 008 ON
16. Close Cascade on XL V 006
17. Put setpoint of 0.1 m on XL L 001
18. Close Cascade on XL V 001
19. Put setpoint of 12.5 gr/s on XL F 001
20. Close Cascade on XL V 002
21. Put setpoint of 2.0 kg/s on XL F 002

Wait until the Liquid Circulation reaches steady state

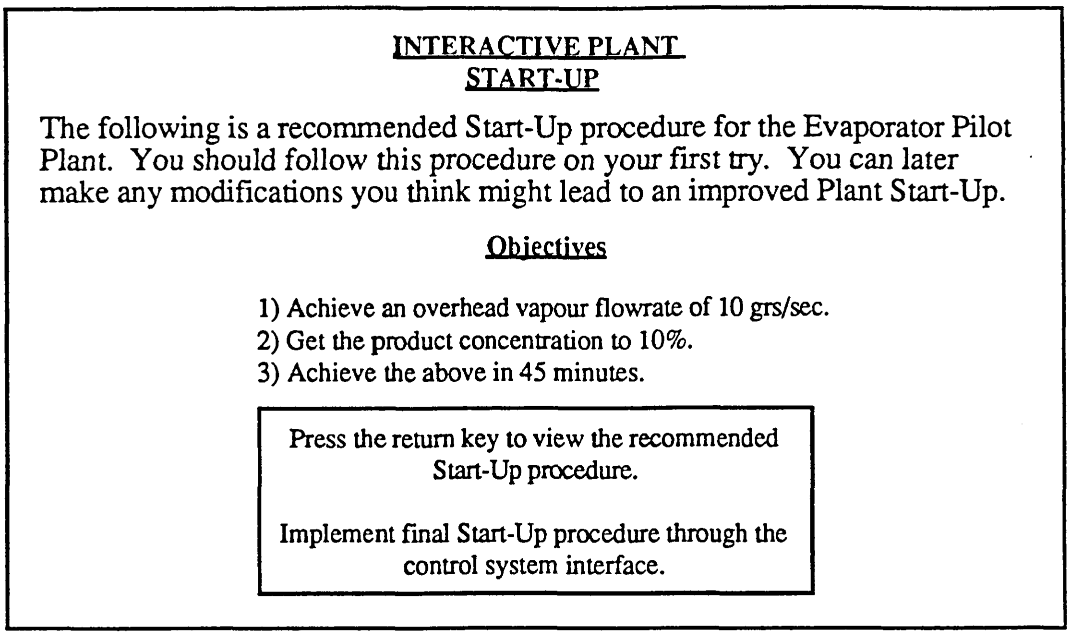
INTRODUCE THE STEAM

22. OPEN XL MV 033
23. OPEN XL MV 037
24. OPEN XL MV 038
25. OPEN XL MV 040
26. OPEN XL MV 043
27. OPEN XL V 003 (20%)

Wait until the Steam Flow stabilizes

28. Close Cascade on XL V 003
29. Put setpoint of 20.0 gr/s on XL F 003

Figure 5.8. An example start-up sequence that achieves the operational objectives outlined in figure 5.9.



INTERACTIVE PLANT
START-UP

The following is a recommended Start-Up procedure for the Evaporator Pilot Plant. You should follow this procedure on your first try. You can later make any modifications you think might lead to an improved Plant Start-Up.

Objectives

1) Achieve an overhead vapour flowrate of 10 grs/sec.
2) Get the product concentration to 10%.
3) Achieve the above in 45 minutes.

Press the return key to view the recommended Start-Up procedure.

Implement final Start-Up procedure through the control system interface.

Figure 5.9. An example display setting out the operational objectives for fully interactive start-up

The instructional process is based on the same principles as those followed in the *Tutorial.* In summary, tasks of low complexity are examined first, then higher order tasks, etc. Furthermore, similar to the *Tutorial* different instructional approaches are used for engineer and operator training.

A second difference arises from the need for engineers to be able to devise their own operational sequences. In the interactive mode while operators are expected to implement a particular sequence as it is given to them, engineers are encouraged to develop and implement their own sequences. Initially engineers are provided with a suggested sequence that achieves the operational objectives which they may implement to gain experience. However, they are then encouraged to devise and implement an improved procedure that achieves the operational objectives with more efficiency.

Finally, the evaluation of the operators is based on the efficiency in which they have implemented the suggested sequences as well as the ability to explain the reasons behind particular sequences. For the engineers the same criteria are examined along with an additional factor arising from the effectiveness of their own sequences. These are evaluated according to the time taken to achieve the operational goals, the plant stability when their sequence is being implemented, etc.

5.4. Training for Fault Identification

One of the important aspects of the operator's role is to deal with faults that upset the plant's normal operation and so minimize any associated plant or unit shut-downs and also reduce the likelihood of damage to plant and personnel. However, fault management is a broad description of an ability that encompasses several activities which involve different skills. Shepherd (1986) identified these skills as: 'detection'; 'diagnosis'; 'compensation'; 'rectification'; and 'recovery'.

The detection of an abnormal situation is the first step in dealing with faults. Often alarms caution the operator of undesirable or abnormal operating conditions. However, a general ability to infer the state of the process from the information given on the interface is necessary. This is best exemplified in situations when the interface does not give complete information. For example, a process interface does not show directly that there is a leak in a part of the process, this is inferred by comparing data about flows and/or levels.

Once the operator has detected an abnormality in plant operation the next step is to identify its cause. Operators are often assisted in this fault diagnosis task by decision support systems, decision trees, cause and effect chains where they can trace back to identify the cause, etc. However, as discussed in chapter 2 these methods have several limitations most prominently indicated in situations where the fault that has occurred has not been foreseen. In these situations an operator has to perform the diagnosis unaided. It is therefore indicated that the operators who are expected to fulfil this task need "to acquire some kind of versatile strategy enabling diagnosis of unfamiliar faults" (Shepherd, 1986).

Compensation is normally associated with maintaining a major plant function until the cause of the abnormal plant behaviour has been identified and/or rectification has been completed. Therefore, operators need to be familiar with the different methods indicated for maintaining a certain function (for example, the cooling of a reactor system). Alternatively, if the maintenance of a function is not possible in view of the prevailing situation, operators need to be familiar with safe intermediate process states and be able to attain these safely and efficiently.

The rectification of an operational problem is usually obvious when the problem has been identified and it often involves the implementation of predefined procedures. Thus the operator responsible for rectifying a fault needs to be able to implement these sequences of operations.

Finally, once the fault has been rectified the operator needs to bring the process back to its normal operational state. This recovery task "is another instance where the operator must select a possible course of action and act on it. Sometimes the course of action is a familiar one and requires no reflection by the operator. Sometimes he must work out the consequences of the action he has chosen and decide whether or not it should be carried out or some alternative sought" (Shepherd, 1986). Therefore, the operator must have adequate process knowledge which will enable him to select the most appropriate course of action.

As noted above the different skills involved in fault handling are generally dependent on different knowledge and experiences and as such demand different training schemes to be conveyed to the process operator. In this section the general structure of a training programme is described that provides the platform for practising fault detection and also aims to convey to the operator a general fault identification strategy. Finally, the application of this programme in a particular case study using the facilities offered by the ISA (presented in chapter 4) is described in chapter 7.

Fault Identification Training Session

Fault identification involves a type of inference which is the inverse of that promoted in the two previous training programmes. It is concerned with knowing the effect and working out the cause, rather than the cause and effect path previously described. Therefore, in order for process operators to be able to perform the whole range of their tasks effectively they need to acquire experience in both reasoning paths. This is required by the notion that "if someone can reason in one direction, this does not automatically mean that they can reason in the other direction" (Bainbridge, 1990).

Traditionally, two methods have been used for training operators in fault diagnosis. The first method involves the presentation of conventional theory and consists generally of the material covered by the two training programmes previously described. The second method presents students with a set of diagnostic rules (some general, some plant specific) which they can use in order to identify faults.

Several studies (for example Shepherd et al. (1977) and Patrick and Haines (1988)) have examined the effectiveness of the two methods. While both studies indicated that the use of diagnostic rules improves the operators fault identification skills there is some disagreement on the usefulness of theoretical instruction alone. However, both found that the use of diagnostic rules supplemented with context-specific theoretical instruction was the most efficient approach.

The fault diagnosis methodology that forms the basis for this training programme is a result of a two years experience in supervising the operation of a CO_2 Absorption/Desorption pilot plant by second year undergraduates in the department of chemical engineering of Imperial College. While no formal evaluation of the effectiveness of this method has been performed, my personal experience is that students presented with the reasoning approach during operation were better able to deal with the numerous plant disturbances as compared to other students.

The method assumes that the student has already had some experience with running the plant and also has the general background for plant operation. Thus, this training programme complements the two previously described by using the knowledge and experiences gained by the operator in those settings to develop and practice a fault diagnosis strategy. Furthermore, the method is based on the execution of a sequence of steps, some procedural and some in the form of diagnostic rules. These are listed in table 5.5.

Table 5.5. Sequence of steps for fault diagnosis

1. Detect abnormal behaviour.
2. Locate the problem area(s) according to the plant subsystems.
3. If more than one subsystems is affected then focus on the one upstream unless there is a recycle connection between the two.
4. Find which variables in the subsystem have been affected.
5. Based on cause and effect relationships already studied identify the inputs and outputs to the medium level unit of the subsystem.
6. If there is a problem with an input to a medium level unit and the subsystem upstream does not show abnormal behaviour examine the low level units that affect that input.
7. If there is a problem with a medium level unit output only, examine the low level units that affect it. If there is no problem with them examine the medium level unit itself.
8. If there is a recycle between two affected subsystems first examine the inputs to the one upstream then the one furthest downstream, then the one before, etc.

The first step in the sequence for identifying a fault is the detection of abnormal behaviour. There is no provision of formal training for fault detection in this training programme. Instead it relies on the 'hands-on' experience in plant operation gained from the previous training sessions and a set of demonstrations of fault scenarios.

The second step is the location of the problem area. Considering the whole set of variables that describe the status of the process to an operator, it is possible to describe variables as belonging to one or more isolated subsystems (for example, flowrates, temperatures, etc.). Thus, in most situations it is possible to employ a process of elimination based on the set of affected variables and isolate the problem areas. It is further pointed out that at this point not all process variables belonging to each subsystems need to be examined.

Once the problem areas have generally been isolated the operator has further focused his search domain. The third step employs a diagnostic rule which is again dependent on the topological structure of the plant. In most situations effects on a unit downstream are caused by problems to units upstream. Therefore, one should first focus attention on the problem subsystem furthest upstream.

This condition is violated in two cases. First it may be possible that more than one faults are in place. For example, in a situation of two independent faults in two different subsystems, two distinct fault identification problems are formulated which can individually be examined by employing the described method. The second case where this condition is violated is when a recycle forms a closed loop with the first and last subsystems which have been identified as problem areas (described as step 8). In this event one should first examine the state of the inputs to the first subsystem. If no problem is identified then the subsystem from where the recycle originates should be examined, then the one before, etc.

All the steps previously described can be performed with a partial knowledge of the whole set of affected variables since the main objective is to focus the search to potential problem areas. The fourth step involves the identification of all the important variables that have been affected within the subsystem.

Based on the cause and effect relationships already studied in previous training sessions an operator should be able to determine (step 5) the inputs and the outputs to the main unit in the subsystem and possibly have a 'feel' on whether their respective measurements are representative of the process state.

Then if a problem is identified with an input to the medium level unit of this furthest upstream subsystem the operator can focus his attention on what may actually be the cause of this deviation. This would involve the examination of the low level units that affect that particular input (step 6).

If however, there is no problem with the medium level unit's inputs while there is a problem with one or more of its outputs the focus of attention should be directed at the low level units that affect them (step 7). If there is no problem with them the operator should examine the operation of the medium level unit itself.

The general format of the training programme is similar to that of the Plant Operation Training Session described in the previous section. Here again there is a demonstration and an interactive mode serving similar purposes. In the demonstration mode the student is presented with the effects of a variety of faults on the status of the process through the use of the training simulator facility. A student may choose (engineer) or be directed (operator) to observe the effects of a range of single or multiple simultaneous disturbances.

As in the demonstration of operational sequences described in the previous section qualitative information is used to emphasize important events during the course of a fault scenario demonstration. An extension of the demonstration capabilities in this session is the ability to show how the fault diagnosis strategy may be used. In this case once a fault occurs the application of the strategy is demonstrated. First the affected variables are indicated, then the reasoning for isolating problem areas, etc.

In the interactive mode the student is presented with fault scenarios of varying difficulty. Similar to the case above, he may be directed or choose which one to study. However, the student is not given any indication of either the nature of the fault or the degree of difficulty associated with it. Once a fault scenario is initiated there are two aspects in the instructional process.

The first is the detection of abnormal plant behaviour and the rectification of the fault. For example if the problem has been caused by the closure of a certain manual valve, the operator is given a prespecified amount of time to realize that normal operation has been violated, diagnose the fault and rectify it, in this case by opening the manual valve. Having done this the operator is informed that the fault scenario has been effectively dealt with and he is given reinforcing feedback to enhance the educational value. Then the instructional process continues with the initiation of a new fault scenario.

The second aspect in the instructional process comes into play if the operator has not managed to rectify the fault in the given time even if he has detected abnormal plant behaviour. In this instance the subsequent instruction takes the form of a dialogue which is carried out with the student. The dialogue follows the *Socratic method* and is based on the problem solving strategy described above (an example dialogue taken from the application described in chapter 7 is given in figures 5.10.).

First, the student is asked to identify the affected variables. Then his list is supplemented with any variables that he may have missed and the new list is presented to him. The next stage of the dialogue asks the student to indicate the problem area(s). If the student correctly identifies the problem area then he is provided with reinforcing feedback and the dialogue proceeds to the next step of the strategy. If however, the student does not identify the problem area correctly he is provided with feedback about why his choice was wrong and hints that aim to help him in making a more accurate decision.

TUTOR: *Abnormal plant behaviour has been detected. Check the status of the process and identify the variables which exhibit deviation from their normal value.*

STUDENT: XL F 005, XL T 002, XL T 010

TUTOR: *You have correctly identified three of the measurements that have been affected. A fourth measurement is XL V 004. Therefore, the whole set of affected variables is:*

XL F 005
XL T 002
XL T 010
XL V 004

Based on this information identify the problem subsystem.

STUDENT: THE EVAPORATOR SUBSYSTEM

TUTOR: *It is true that a problem in the Evaporator Subsystem could affect some of the measurements previously identified.*

In particular XL T 002 and/or XL T 010. However, we have seen that also XL F 005 and XL V 004 have been affected.

Since the control loop between XL F 005 and XL V 004 is a simple flow controller not affected by other parts of the plant the problem area cannot be the Evaporator Subsystem.

Try Again.

STUDENT: THE CONDENSER SUBSYSTEM

TUTOR: *You have correctly identified that the problem is in the CONDENSER SUBSYSTEM.*

Has an input to the Condenser deviated from its normal value?

STUDENT: NO

TUTOR: *We have seen that both XL F 005 and XL V 004 have deviated. XL F 005 represents the cooling water input and is regulated by the opening of the XL V 004 control valve.*

Has an input to the Condenser deviated from its normal value?

STUDENT: YES

TUTOR: ***Which one(s)?***

STUDENT: XL F 005, XL V 004

TUTOR: *That is correct.*

Let us examine the effects.

XL T 002 has reached almost the value of XL T 007
XL T 010 has decreased to almost the value of XL T 009
XL F 005 has gone to zero
XL V 004 is OPEN 100%

What would the combination of these effects indicate?

Figure 5.10.a. A example dialogue of the *Fault Identification Training Session.*

STUDENT: NO COOLING WATER

TUTOR: *That is correct.*

Check the setpoint of XL F 005.

Is it equal to zero?

STUDENT: NO

TUTOR: *That is correct.*

Furthermore, since XL V 004 has OPENED 100% to accommodate for the lack of cooling water flow, we may conclude that there is no problem with the control loop.

Examine the low level units that could stop the cooling water flow.

Is there a problem with any of them?

STUDENT: YES

TUTOR: ***Which one?***

STUDENT: XL MV 047

TUTOR: ***What is the problem?***

STUDENT: XL MV 047 IS CLOSED

TUTOR: *That is correct.*

Is there a problem with any other ones?

STUDENT: NO

TUTOR: *That is correct.*

So what was the problem in this fault scenario?

STUDENT: XL MV 047 IS CLOSED

TUTOR: *That is correct.*

You have correctly identified the problem. Indeed when any of the manual valves that regulate the cooling water supply is CLOSED then there is NO COOLING WATER SUPPLY to the subsystem.

In summary:

1. *You identified that the problem was in the CONDENSER SUBSYSTEM*
2. *Then based on the cause and effect information you identified that there was NO COOLING WATER.*
3. *Finally, you identified that XL MV 046 was CLOSED*

Figure 5.10.b. Part B of the example dialogue.

Having identified the problem area the dialogue proceeds in a similar fashion through all the steps of the strategy until the fault is identified. At every level a student is provided with either explanatory or guiding feedback. Finally when the student successfully diagnoses the fault he may continue the instructional process with the study of a new fault scenario.

It is pointed out that the student is not given the choice of identifying the fault until the last step of the strategy has been completed. The reason for this is that the main objective of the session is to convey the general methodology to the student, rather than teach him to diagnose specific faults. The consistent presentation of the methodology aims to achieve that objective. Last, the evaluation of the student in this session depends on his ability to employ the strategy in fault scenarios involving novel faults of the same categories as those studied in the two modes.

5.5. Summary

An examination of the range of functionalities offered by the ISA and the identification of the different training needs for plant operators led to the development of a set of training programmes that aims to provide the trainee with a complete mental model of the plant, the process, the process dynamics and the cause and effect relationships that define plant operation. Three programmes have been developed each addressing a different training objective. These are: (i) the *Tutorial*, (ii) the *Plant Operation Training Session,* and (iii) *the Fault Identification Training Session.*

First, the *Tutorial* introduces the trainee to the various plant components and their characteristics. Then, it builds on this knowledge to introduce the concept of plant subsystems and their respective dynamics and cause and effect relationships. Finally, subsystem interactions are examined until ultimately the whole plant is examined.

The *Plant Operation Training Session* is the second training programme. The emphasis here is on real-time operation. A demonstration mode permits focusing attention on important cause and effect relationships, as well as the presentation of formal operating procedures such as start-up and shut-down. Alternatively an interactive mode allows for actual 'hands-on' training in a real environment. Under this mode the trainee is presented with a set of objectives which have to be met. He can run the simulation in real-time through standard control system interfaces, or he may formalise the definition of complex procedures for start-up and shut-down, etc.

The third training programme aims to teach the trainee a fault identification strategy. When this *Fault Identification Training Session* is used within the context of the whole set of training programmes it also helps to re-enforce the mental model formulated in the two previous ones.

The instructional material presented in these training programmes range from descriptions of the functionality of individual plant components to training for fault diagnosis in a real environment. The instruction is performed through the provision of qualitative explanations, guidance and advice while steady state and real-time simulations provide the means for acquiring 'hands-on' experience of plant operation.

Chapter 6

PROCESS TRAINER: FUNCTIONAL DEFINITION

This chapter describes the implementation of the three training programs in one integrated CBT system which utilizes the entire ISA and which can be easily adapted to provide training for a range of problem areas. The general architecture and the broad characteristics of this system were described in section 4.2.4. while the characteristics of the modules that are employed by this architecture and their respective interconnections where examined in detail in section 4.1.

In particular *Process Trainer* is presented which acts as the intelligent module in this system architecture and is the generic implementation of the training programmes described in the previous chapter. *Process Trainer* forms the basis of the instructional process and uses the simulation based training system (the simulation, control system, and operations management modules) as required by each training session.

Process Trainer has been implemented through several rule-based expert systems which were developed using LEVEL5 (see section 4.1.4.). Therefore, its examination will be performed in three stages. First, the different types of general rules that form the building blocks for the expert systems will be described, followed by the different ways in which the domain knowledge has been represented. Following that, the application of these rules and the use of the domain knowledge within the general structure of *Process Trainer* are presented. Finally *Process Trainer's* adaptation to a particular case study will be described in chapter 7.

6.1. General Rules

Process Trainer has been implemented as a collection of rule-based expert systems. In order to formalize and have a consistency in the development of the expert systems' knowledge bases and achieve general applicability and flexibility, a set of generic rules (listed in table 6.1.) were devised to be used as building blocks for the definition of the different training programmes.

Table 6.1. The generic rules used for the formulation of *Process Trainer*

1. Structural Rules	Permit the definition of formal sequences that the system will delegate and have to be fulfiled before the programme continues with its execution. These provide the skeleton for the training programme.
2. Multiple Choice Rules	Present the student with a set of alternatives out of which he has to select one or more. These can be used to select instructional modes, training situations, etc. as well as select an answer to a multiple choice question.
3. Communication Rules	Handle the communications paths. These can be used to initiate new knowledge bases or to communicate with external programs.
4. Simulation Rules	Permit the definition of simple steady state simulation models within the infrastructure of *Process Trainer.*
5. Diagnostic Rules	Allow the implementation of diagnostic heuristics that may be based on either qualitative or quantitative reasoning techniques.
6. End Rules	Serve as the end point in a procedural tree. They may involve the presentation of concluding statements and/or direct the program to the beginning or to other separate branches of a the procedural tree.

The general structure of the rules is defined by the use of LEVEL5's own Production Rule Language (PRL) and the specific format in which LEVEL5's functionalities (for example communication with external programs) can be exploited. As described in section 4.1.4. in PRL knowledge is represented as IF...AND...OR...THEN...ELSE rules which contain the factual information comprising the domain of the expert systems. However, based on this general structure and using the general format for exploiting LEVEL5's functionalities a set of generic rules have been formulated that can be used for specific purposes within a CBT programme.

The first type of rules which is used in the definition of a training programme is defined as *structural rules*. Their main function is to implicitly control the instructional process. Thus, they form the backbone of the training program and are generally not visible to the student user of *Process Trainer*. As such they permit the definition of formal sequences (for example of other rules of any type) which will be executed once the rule has been triggered. There are two major categories of *structural rules*.

The first category represents rules which are simply *procedural rules* and indicate the order in which a certain sequence of steps will be executed. For example, in reference to the introductory session of the *Tutorial* a student operator is directed to study certain subjects in a predefined sequence. First, he will be introduced to the process, then he will be presented with the plant structure and then he will be directed to study the low and medium level units in the order chosen by the instructor. This rule is given in table 6.2.

The second category of *structural rules* can be described as *evaluation rules* as they use the result of a student interaction in order to initiate an appropriate sequence of steps. For example, at the same point in the *Tutorial* a student engineer is given the choice of which unit to study. The type of rule that enables this selection will be examined later, however a *structural rule* which uses the students choice recalls the relevant instructional material and/or sequence. Such a rule is given in table 6.3. Here a student engineer has selected to study the second low level unit. Thus the rule first presents the instructional material and then initiates the evaluation session for that particular unit.

Table 6.2. An example of a *procedural rule*

RULE	**Introductory Session of the *Tutorial* for student Operators**
IF	Introduce the Process
AND	Introduce the Plant
AND	Present the instructional material for the first low level unit
AND	Present the instructional material for the second low level unit
. . .	
AND	Present the instructional material for the first medium level unit
AND	Present the instructional material for the second medium level unit
. . .	
AND	Perform the Introductory Session Cumulative Evaluation
THEN	The Introductory Session for Student Operators has been completed

Table 6.3. An example of an *evaluation rule*

RULE	**Instruction of second low level unit for student engineers**
IF	Student Choice of unit to study IS Study the second low level unit
AND	Present the instructional material for the second low level unit
AND	Evaluate student's understanding of the second low level unit
THEN	The Student Engineer has studied the second low level unit

The second type of rules that are used for the definition of a training programme are *multiple choice rules*. These present the student with a set of alternatives out of which he has to select one or more. The same format can be used to define all instances where the student has to indicate a choice. This type of rule allow~~s~~ the student to specify the training mode (engineer/operator), choose the next step in the instructional process (review instructional material or continue with subject A), select the actual sequence in which the material may be studied (study the second low level unit, then the third medium level unit), select the correct answer(s) of a multiple choice question, etc.

Once again two categories of *multiple choice rules* can be described depending on the number of choices that a student can make. The first category are *single-answer rules* and allow the student to indicate only one alternative while the second category of *multiple-answer rules* allow the student to make several choices. In most cases the first category of *multiple choice rules* is used.

There are only two instances were the second category is used. The first instance is during the assessment of a student's understanding of a particular topic. In this case the answer to a particular question may be a combination of the answers presented. The second instance where this category is used is during the study of the simulation problems examined in *Process Trainer.* In this case the student can choose to manipulate several inputs synchronously and examine their overall effect on the process or the part of the process that is being simulated.

An example of a *single-answer rule* is given in table 6.4. This rule is taken from the intermediate session for engineer students of the *Tutorial.* At this instance the student engineer has studied the qualitative material that describe a plant subsystem and he is asked to indicate the next step in the instructional path. He may choose to study the steady state or the dynamic operation of the plant subsystem, review the qualitative material for the plant subsystem, review the instructional material of the introductory session (low and medium level units) that relate to this plant subsystem or he may choose to finish with the study of this particular plant subsystem. In this case the student can make only one choice.

Table 6.5. represents a *multiple-answer rule* that allows the student to change one or more of the inputs to a steady state simulation of a plant subsystem. In this case he may choose to change the status of one or more low level unit, the value of one or more process input, or choose to end the study of the steady state operation of this plant subsystem.

Table 6.4. An example of a *single-answer rule*

RULE	**Selection subtopic of subsystem 1 for study, student engineers**
IF	Subsystem 1 subtopic **IS** Steady State Operation of Plant Subsystem
OR	Subsystem 1 subtopic **IS** Dynamic Operation of Plant Subsystem
OR	Subsystem 1 subtopic **IS** Review Qualitative Description of Plant Subsystem
OR	Subsystem 1 subtopic **IS** Review Introductory Material relating to the Subsystem
OR	Subsystem 1 subtopic **IS** End the Study of this Subsystem
AND	Execute the student's choice
AND	Evaluate student's understanding of the plant subsystem 1
THEN	The Student Engineer has studied the plant subsystem 1

Table 6.5. An example of a *multiple-answer rule*

RULE	**Change of inputs to steady state simulation of subsystem 1**	
IF	Subsystem 1 change input **ARE**	Status of first low level unit
OR	Subsystem 1 change input **ARE**	Status of second low level unit
•		
•		
•		
OR	Subsystem 1 change input **ARE**	Value of process input 1
OR	Subsystem 1 change input **ARE**	Value of process input 2
•		
•		
•		
OR	Subsystem 1 change input **ARE**	End Study of Subsystem 1 Steady State
AND	Execute the Steady State simulation of Plant Subsystem 1	
THEN	The Student Engineer has studied the plant subsystem 1	

Communication rules are the third type of rules that are used in *Process Trainer*. These rules enable the definition of the communications paths and thus make the network used by *Process Trainer* possible. They can be used to initiate new knowledge bases or to communicate with external programs, in this case to communicate with the simulation based training system (SBTS). There are again two main categories associated with these rules.

It was mentioned above that *Process Trainer* is made up of a number of knowledge bases that can be executed independently. Therefore, in order to establish this network of knowledge bases it is necessary to be able to indicate which one should be active in a particular instance. The first category of *internal communication rules* addresses this need. When such a rule is triggered then it activates a different knowledge base.

For example, when a student begins the instructional process with *Process Trainer* he is given the choice to select one of the three training session that were described in the previous chapter *(the Tutorial, the Plant Operation Training Session, or the Fault Identification Training Session).* These training programmes are described in different knowledge bases. Therefore, the selection of one of the programmes from the main *Process Trainer* knowledge base activates the knowledge base that relates to that particular programme (i.e. if the *Tutorial* is selected then the *Tutorial* knowledge base is activated etc.). A rule representative of this category is shown in table 6.6.

Table 6.6. A *communications rule* which activates a different knowledge base.

RULE	**Start the *Tutorial* Training Session**
IF	The Selected Training Session **IS** The *Tutorial*
THEN	The Student has selected the *Tutorial* Training Session
AND	**CHAIN** The *Tutorial* Knowledge Base

A second category of *external communication rules* allows the utilization of the SBTS as an instructional tool. These rules are able to activate specific executable files which are present in the disk-drives of the computer where LEVEL5 resides. Furthermore, it is possible to issue commands specific to the external program and send parameters to these programs from the rules. Examples of *external communication rules* are shown in tables 6.7.

Table 6.7.a. gives the format used for the communication with the modules of the SBTS. Here the executable file includes a sequence of DOS commands that are able to communicate with each of the SBTS modules through the PC3270 emulation software (see section 4.3.). These latter commands may initiate a simulation problem, may send data and direct changes to the control system module (setpoint changes, status changes, etc.), set up the appropriate operations management module files that correspond to a particular problem, or receive data from these modules.

A different level of communication is achieved with the rule presented in table 6.7.b. This rule is representative of the rules used by *Process Trainer* to recall graphics created with Microsoft Paintbrush at different stages in the instructional process. In this rule we see that first there are two components to the communications statement. The first component again activates the external program (in this case PAINT.EXE) which permits the incorporation of the 'picture' as part of the knowledge base. The second component (COMMAND) determines which picture file should be retrieved from the computer's hard-disk.

Table 6.7.c. shows the format for the communication with an external program which requires certain parameter inputs and returns to the knowledge base a set of parameters that have been evaluated by the external program.

For example the external program could be a PASCAL program that receives a set of inputs, which have been determined from the student's interaction with the knowledge base, and then returns certain evaluated parameters relevant to the knowledge base session. However, it is also possible to activate an external program which takes its inputs from a different source and then returns to the knowledge base a set of desired parameters. In *Process Trainer* this approach is used to activate a PASCAL program that reads the current time and date and returns them to the knowledge base. In this example the PASCAL program accesses the computer clock and then returns the required values in a format appropriate for use in *Process Trainer*.

Table 6.7.a. A *communications rule* for linking *Process Trainer* to a module of the Simulation Based Training System.

RULE **Link with a module of the SBTS**
ACTIVATE <path> filename.bat
THEN Command file has been issued

Types of filename.bat files

1. Initiating either the Simulation or the Operations Management Module

```
Send second_filename.bat filename $Command a (crlf ascii)
```

   a) Second_filename.bat for Simulation Module

```
smsg account_name simulation_problem_exec
```

   simulation_problem_exec: *STARTUP EXEC*

   STARTUP EXEC: *SPEEDD XLSTART*

   b) Second_filename.bat for Operations Management Module

```
smsg account_name operations_management_view_exec
```

   operations_management_view_exec: XLSTARTUP.EXEC

   XLSTARTUP.EXEC: *SPRBTCH xlstartmp xlstartop xlstartrt xlstartpd*

2. Communication with RPTMS

   a) Sending Data

```
Send filename.lst filename $acsdata a (crlf ascii)
Send filename.dat filename $acsdata a (crlf ascii)
```

   b) Requesting Data

```
Send filename.lst filename $acsdata a (crlf ascii)
Receive filename.dat filename $acsdata a (crlf ascii)
```

Filename.lst format		Example from RPTMS	
ACS	*CLASS*	*ACS*	*CLASS*
Variable1_name	*Variable1_item-number*	*XLV1*	*10*
Variable2_name	*Variable2_item-number*	*XLMV33*	*10*
.	.	.	.
.	.	.	.
.	.	.	.
VariableN_name	*VariableN_item-number*	*XLF3*	*40*

Filename.dat format		Example from RPTMS	
N		*N*	
Variable1_Type	*Variable1_value*	*R*	*33.333*
.	.	.	.
.	.	.	.
.	.	.	.
VariableN_Type	*VariableN_value*	*R*	*20.0*

Where Variable_Type could be Real (R) or Integer (I)

Table 6.7.b. A *communications rule* for recalling a graphics image

RULE	**Recall the first picture describing a heat exchanger**
ACTIVATE	<path> Paint.exe COMMAND Heat_Exchanger1.pcx
THEN	First picture describing a Heat Exchanger has been recalled

Table 6.7.c. A *communications rule* for accessing an external program

RULE	**External program activation (inputs and outputs)**
ACTIVATE	<path> External_Program.com
SEND	Parameter 1
SEND	Parameter 2
•	
•	
•	
SEND	Parameter N
RETURN	Parameter 1
RETURN	Parameter 2
•	
•	
•	
RETURN	Parameter J
THEN	External program has been activated

Finally, the case where the knowledge base requires as input a set of data that has been written in an external file is considered. This data file could have been created previously by the training developer or it may have been created dynamically as a result of the execution of an external program. For example in the case which *Process Trainer* requests data from RTPMS it was seen in table 6.7.a. that the requested data are received as a data file. Thus a different rule is required which will enable *Process Trainer* to access these data. The general format of these rules is presented in table 6.7.d.

The rule types which have been described up to now enable the formal definition of the skeleton of the training programmes and also the setting up of the communication network of relevant knowledge bases and the use of the simulation based training system from different points of *Process Trainer*. However, a different set of functional rules add substance to the training programmes.

Table 6.7.d. A *communications rule* for reading an external data file

```
RULE  For reading the data from an external data file
READ  <path> filename.dat
DATA  Parameter 1
DATA  Parameter 2
   .
   .
   .
DATA  Parameter N
THEN  Resume processing
```

The first type of these rules are *simulation rules*. These permit the definition of simple sequential steady state simulation models within the infrastructure of *Process Trainer* itself (that is, in LEVEL5, not in the external simulation module). These rules will take as inputs data resulting from student responses or data that have been received from an external program (possibly as a data file). The structure of these rules is given table 6.8. Here it is seen that the simulation model consists of a sequence of equations that are processed sequentially when the rule is triggered.

Furthermore, it is possible to include conditional statements which determine the actual sequence in which these equations are solved. In the example described the inputs to the rule come from other parts of the knowledge base. Initially these are first used to determine which sequence of equations to process, then they are used to evaluate one function, then this result is used as an input to a new function etc.

Another type of rules which are used in *Process Trainer* are *diagnostic rules*. These rules permit the implementation of diagnostic heuristics within the structure of the training programmes. Furthermore, the heuristics could be based on either qualitative or quantitative reasoning techniques, or even a combination of the two.

Table 6.8. A *simulation rule*

```
RULE  Steady state simulation model of medium level unit 1
  IF  Manual Valve 1 = Closed
 AND  m > n
 AND  F = f(m,p,q,...)
 AND  G = f(F,C,... )
   •
   •
   •
  OR  Manual Valve = Open
 AND  m > n
 AND   H = h(m,n,...)
   •
   •
   •
  OR  Manual Valve = Open
 AND  m < n
 AND  J = j(n, p, q,...)
   •
   •
   •
 AND  Present the results of the steady state simulation of the medium unit 1
THEN  Steady state simulation of medium unit 1 has been processed
```

These rules are of particular relevance to the detection of abnormal plant behaviour and the diagnosis of process faults. In addition these rules form the basis for the evaluation of a students response to a question during an assessment session. The general format of the diagnostic rules is given in table 6.9.

Finally, the last type of rules that are used in *Process Trainer* are *end rules*. These rules serve as the end points of a branch of a procedural tree. They may involve the presentation of concluding statements or they may direct the program to the beginning or to other separate branches of the procedural tree. For example, let us again consider the example of the student engineer in the introductory session of the *Tutorial* who has selected to study a specific unit. Once he has finished studying that unit he should be taken back to the step where he can select to study a new unit.

Table 6.9. The format of *diagnostic rules*

```
RULE  Diagnostic Rule for fault...
  IF  String Condition (1,1)
  OR  String Condition (1,2)
   •
   •
   •
  OR  String Condition (1,N)
  OR  Numeric Condition (1,1)
  OR  Numeric Condition (1,2)
   •
   •
   •
  OR  Numeric Condition (1,N)
 AND  String Condition (2,1)
 AND  String Condition (2,2)
   •
   •
   •
 AND  String Condition (2,N)
 AND  Numeric Condition (2,1)
 AND  Numeric Condition (2,2)
   •
   •
   •
 AND  Numeric Condition (2,N)
   •
   •
   •
 AND  String Condition (M,N)
 AND  Numeric Condition (M,N)
THEN  Process Fault is Fault...
```

There are two steps in the execution of this rule. Since the knowledge base, being dynamically updated, 'knows' all the results of a current session the first step to an *end rule* is to direct the knowledge base to 'forget' any data that will inhibit the effective repetition of a particular sequence. For example, the knowledge base 'knows' that the student chose to study a particular unit.

Therefore, unless that fact is forgotten the repetition of the *multiple-choice rule* that allows the selection of a unit for study will not be treated as an interactive rule, since the answer is already available to the program. Thus the previous selection ought to be forgotten so that the rule requests a selection to be made by the student. The second step is to loop back to the beginning of that particular rule. An example of an *end rule* is given in table 6.10.a.

Table 6.10.a. An *end rule* which repeats a procedural tree or branch

```
RULE  Redirecting student engineer to select a new unit for study
  IF  The Student Engineer has studied the first low level unit
  OR  The Student Engineer has studied the second low level unit
   .
   .
   .
  OR  The Student Engineer has studied the last low level unit
  OR  The Student Engineer has studied the first medium level unit
  OR  The Student Engineer has studied the second medium level unit
   .
   .
   .
  OR  The Student Engineer has studied the last medium level unit
THEN  Repeat the unit selection rule
 AND  FORGET ALL
 AND  LOOP
```

Above the way in which an *end rule* may be used to loop back to the beginning of a procedural tree or branch was described. Alternatively an *end rule* may be used to direct the knowledge base to continue its processing at a different point of the general procedural tree. In this case a *structural rule* acts as an *end rule.* For example in table 6.10.b. a rule is given that uses the fact that the student has successfully answered all the questions relating to the introductory session of the *Tutorial* to take him to the beginning of the intermediate session.

Finally it is pointed out that the basic format of the general rules described above can be enhanced in order to exploit further functionalities offered by LEVEL5. An "ELSE" statement may be used to assert an alternate conclusion when the failure of a rule's primary conclusion supports the alternate conclusion. Also the "FILE" PRL command may be included in any of the rules in order to record a student's interaction and path through the material to an external text file. Then this external file can be viewed at a later time by the trainer and/or the student so that a collective assessment of the training session can be performed.

Table 6.10.b. A *structural rule* as an *end rule* which directs the knowledge base execution to a new a procedural tree or branch

RULE	**Redirecting a student engineer to select a new unit for study**
IF	The Student Engineer has successfully completed the introductory session
AND	Begin the intermediate session for student engineers
THEN	End the introductory training session

Last the use of the "DISPLAY", "TEXT", and "EXPAND" PRL commands enrich the student interaction with the training programmes through the provision of supplementary information.

6.2. The Domain Representation

Having examined the different types of rules that one can use to build a rule-based training programme it is necessary to consider how the problem domain which this programme addresses can be represented. Already the different forms in which the domain is represented have been discussed in chapters 4 and 5. These are summarized in table 6.11.

Verbal knowledge representations add substance to the structural skeleton of a rule-based training programme. Furthermore, it is this substance that determines the application of the rule-based expert system, as it is used to define the interactive dialogues, to describe the instructional material, and to provide explanations. In summary, verbal representations of the domain knowledge guide the student through the instructional process and enable him to access the underlying functionalities of *Process Trainer.*

Table 6.11. Forms of knowledge representation

1. Verbal	Verbal knowledge representation is used to define the interactive dialogues with the student, to provide verbal descriptions of the instructional material, and to provide explanations.
2. Diagrams	Non-interactive diagrams provide static descriptions of the instructional material, while interactive diagrams permit the direct interaction of the student with the instructional material.
3. Steady State Problems	Steady state simulations are used to represent the steady state operational characteristics of a process or a process unit.
4. Dynamic Problems	Dynamic simulations are used to represent the dynamic properties and operational characteristics of a process or a process unit.
5. Operational Sequences	These sequences are used to describe operational actions that are characteristic of normal and/or abnormal plant behaviour.
6. Heuristics	Heuristics are used in the detection of abnormal plant behaviour and the identification of faults.

The types of *verbal knowledge representation* that have been used in the training sessions of *Process Trainer* are dependent on the functionalities of LEVEL5. Nevertheless different types of *verbal knowledge representation* have been used in the training sessions of *Process Trainer*. The different types may be categorized as: *transitional statements, descriptive statements,* and *explanatory statements.*

Transitional statements are used to make a training programme cohesive and make the transition from one step of the instructional process to the next more natural. Generally, *transitional statements* are used to: (i) describe the actions that *Process Trainer* expects a student to carry out (for example, "make an input change to a simulation problem", "choose an answer to a multiple-choice question", "get display XL90 on the control system interface", etc.), and (ii) provide information to a student about an action that *Process Trainer* is performing (for example, "initializing the simulation based training system"). These statements are incorporated into the relevant rules of the knowledge base with the "TEXT" and/or "DISPLAY" PRL commands.

Table 6.12.a. is a simple example of a rule that uses the TEXT function. This rule would ask a student to respond to the questions by answering TRUE or FALSE. Here the first text identifier is "Question 1". When this question is posed to the student it is replaced by "Have you attended a tutorial session on the Plant Components of the Pilot Plant?" The student then has to respond to this question before the processing of the knowledge base is resumed.

Table 6.12.a. Use of the TEXT function in *Process Trainer*

```
RULE  For posing questions to a student
  IF  Question 1
 AND  Question 2
  OR  Question 3
   .
   .
   .
THEN  The end of question session

Body of extended text
TEXT  Question 1
        Have you attended a tutorial session which examined the
                 Plant Components of the Pilot Plant
TEXT  Question 2
              Textual Information Relating to Question 2
TEXT  Question 3
              Textual Information Relating to Question 3
   .
   .
   .
```

The DISPLAY function permits the knowledge engineer to designate supplemental information that is automatically shown to the user from a title, goal, supporting condition, or conclusion of the knowledge base. Similar to the use of the

TEXT function, to display text from within a rule, the text is given an identifier which is used to designate from where the information is to be shown.

An example of how this function is used is shown in table 6.12.b. Similar to the TEXT example when the rule executes its first rule (i.e. to present the student with display 1) it substitutes the identifier with the previously provided textual information that correspond to that identifier (i.e. "You have selected to begin the Tutorial Training Session. The main objective of this session is...etc."

Descriptive statements are used in *Process Trainer* for the verbal description of the instructional material. It is possible to consider them as a special case of *transitional statements,* however, since they are used to describe particular subjects they are not dependent on the formal structure of the instructional process. On the contrary they may be retrieved and used as necessary at different points of the knowledge base.

For example, in the description of the *Tutorial* it was mentioned that at different occasions a student is given the option of reviewing instructional material from previous levels. In this case the classification of the verbal descriptions relating to particular subjects as *descriptive statements* in the knowledge base, permits easy and flexible access. Nevertheless, the format used for the definition and use of these statements is the same as that presented in table 6.12.b.

The last category of *verbal knowledge representation* is *explanatory statements*. These are used to provide explanations to a student upon his request at different points of the instructional process. In this way the explanatory statements do not constitute formal steps in a training sequence. Instead they can be used to clarify questions that are posed during the instructional process, describe any options offered to the student or even provide a preview of the sequences that follow.

Table 6.12.b. Use of the DISPLAY function in *Process Trainer*

```
    RULE  For presenting a sequence of textual information to a student
      IF  DISPLAY Display 1
     AND  DISPLAY Display 2
      OR  DISPLAY Display 3
       .
       .
       .
    THEN  The presentation of the information has been completed

Body of Displays with supplemental information
  DISPLAY Display 1
          You have selected to begin the Tutorial Training Session.
                The main objective of this session is...etc.
  DISPLAY Display 2
                        Textual Information
  DISPLAY Display 3
                        Textual Information
       .
       .
       .
```

This is achieved with LEVEL5's "EXPAND" PRL command. This permits the knowledge engineer to designate supplemental information that is shown to the user only when requested by the use of an appropriate function key on the computer keyboard. With the EXPAND function the text is once again given an identifier following the reserved EXPAND and can be accessed from wherever the identifying phrase occurs.

A simple use of the EXPAND function from within a rule is given in table 6.12.c. Here the first condition of the rule asks the student to confirm that a specific control system variable is on alarm while the explanatory statement indicates where and in what form that information is presented.

Table 6.12.c. Use of the EXPAND function in *Process Trainer*

```
    RULE  For providing explanatory information to a student
      IF  The level of the product tank is on alarm
     AND  Condition 2
      OR  Condition 3
       .
       .
       .
    THEN  The end of rule

Body of Expanded text with supplemental information
   EXPAND  The level of the product tank is on alarm
  The variable describing the level of the product tank in the Plant is XLL1.
        This can be found in the control system display XL900...etc.
   EXPAND  Condition 2
              Textual Information relating to Condition 2
   EXPAND  Condition 3
              Textual Information relating to Condition 3
       .
       .
       .
```

Finally, it is pointed out that all three categories of verbal knowledge representation can be used in one rule thus increasing its overall effectiveness. An example rule that uses all three functions described above is given in table 6.12.d. Here the student is asked to select which of the *Tutorial's* training sessions he wants to follow.

The TEXT function is used to pose the question in a natural language format and indicate how the student may indicate his selection. At the same time the EXPAND function offers an explanation and a brief description of each of the training sessions. Thus the student may use this function to preview the three sessions before he makes his final choice. Finally, once the student has made his choice the DISPLAY function is used to confirm his selection.

Table 6.12.d. An example of a *single-answer rule* which utilizes all three categories of *verbal knowledge representation.*

RULE **Selection session to study**
IF Session to be studied **IS** The Tutorial
OR Session to be studied **IS** Plant Operation Training Session
OR Session to be studied **IS** Fault Identification Training Session
AND **DISPLAY** Student's choice of training session
AND Start the selected training session
THEN The Student has selected a training session

Verbal knowledge representations related to this rule

TEXT Session to be studied
Please indicate the training session you want to study.
Use the arrow keys of the keyboard to move the pointer to your selection.
When the pointer is at your choice press the return key to continue.

EXPAND Session to be studied
At this point you are expected to select one of the three training session.
Each one covers a different aspect of plant operation...etc.

DISPLAY Student's choice of training session
You have selected to study the [session to be studied]...etc.
Press return to initiate that training session.

The second form of knowledge representation that is used in *Process Trainer* is the *diagrammatical knowledge representations.* As the name implies this form of knowledge representation uses graphic images to describe the problem domain. The use and the different types of diagrams that have been used in *Process Trainer* as well as representative examples were described in chapter 5. Two categories of diagrams can be identified: *interactive* and *non-interactive.*

The first category of *interactive diagrams* is of particular relevance to the simulation based training system. In particular interactive displays available from the control system module interface permit the student to directly manipulate process variables, perform possible control actions, and schedule operational sequences.

The second category is *non-interactive diagrams*. Two main types are used in *Process Trainer, dynamic* and *static representations. Dynamic representations* are again of particular relevance to the simulation based training system. They represent diagrams available at the control system interface which are dynamically updated. Examples include mimic diagrams which include process variables that change with time, as well as diagrams which describe a process variables dynamic response.

Static representations are used to provide descriptions of the instructional material in the form of mimic diagrams, cause and effect diagrams or show specific transients of process variables. Furthermore, the static representation of a particular situation could combine all different forms outlined above along with textual information.

Most of the *non-interactive diagrams* that have been used in *Process Trainer* were developed outside LEVEL5 with Microsoft Paintbrush. The way these diagrams can be accessed by the training programmes was described in section 6.1. Nevertheless some non-interactive diagrams were developed using the limited graphics capabilities of LEVEL5 and are used with the DISPLAY function described above. In particular these diagrams are used in the steady state simulation studies to show newly achieved steady states which have resulted from student manipulation of the process or process unit inputs.

The types of knowledge representation described above represent different ways which make the presentation of the instructional material and the interaction with the instructional session more natural. However, part of the domain knowledge that needs to be represented for a process plant is functional. For example we have previously described the importance of simulation for the training of operations staff, the need to teach operators the important operational procedures, and the usefulness of heuristic rules for the training of a fault diagnosis strategy. *Process Trainer* uses three different types of *functional knowledge representation.*

The first type is knowledge which describes the operational characteristics of a process or a process unit. This is described in steady state and dynamic simulation models which are then coupled with the appropriate interfaces. In the case of dynamic simulations these are coupled with control system functions to create simulation problems that can be utilised at different points of the instructional process.

The use of *steady state simulation problems* in the instructional process was described in chapter 5. Furthermore, the general format of the *simulation rules* that can be used to define steady state simulation models within *Process Trainer* was presented in section 6.1. These models may comprise one or more rules and are unified by a single controlling rule that determines the sequence of steps involved in the simulation problem. Therefore, it is possible to access a particular simulation problem from any point of the knowledge base just by calling the controlling rule. An example is given in table 6.13.

The use of *dynamic simulation problems* which can describe the dynamic properties and operational characteristics of a process or a process unit in the instructional process was also discussed in chapter 5. Furthermore, the special functionalities and the definition of the dynamic simulation problems which utilize the simulation and the control system modules of the ISA were presented in chapter 4.

Table 6.13. An example of a *steady state simulation model*

```
RULE  Steady State Simulation model
  IF  Rule for getting the necessary input data for this simulation
 AND  Simulation rule 1
 AND  Simulation rule 2
   •
   •
   •
 AND  DISPLAY Simulation results display
THEN  Simulation of steady state model has been completed
```

These problems are stored outside *Process Trainer* knowledge bases but they can be used as required at different points of the instructional process. Finally the way these simulation problems can be accessed from within *Process Trainer* was described in section 6.1. The dynamic simulation model of the evaporator pilot plant is described in chapter 7.

The definition, importance and use of *operational sequences* which describe operational actions that are characteristic of normal or abnormal plant behaviour, in the instructional process have been previously described in chapters 4. and 5. Furthermore, the integration of these sequences, which are defined in the operations management module of the ISA, with the simulation and control system modules was described in chapter 4.

Similar to the dynamic simulation problems *operational sequences* once defined are again stored outside *Process Trainer.* The way these sequences can be accessed from within the knowledge bases has also been described in section 6.1. Finally, examples of models which describe operational sequences for the evaporator pilot plant application are presented in chapter 7.

The last form of domain knowledge that has been represented in *Process Trainer* is *heuristics* which are used for the detection of abnormal plant behaviour and the identification of faults. *Heuristics* may comprise one or more rules that are grouped in a controlling rule, and uses *diagnostic rules* (section 6.1.). As with *steady state simulation problems*, all *heuristics* can be accessed directly from different points of the instructional process. An example controlling rule which implements a *heuristic* is shown in table 6.14.

Table 6.14. An example of a representation of a *heuristic.*

RULE	**Heuristic that detects abnormal plant behaviour**
IF	Receive data describing plant status
AND	Diagnostic rule 1
AND	Diagnostic rule 2
OR	Diagnostic rule 3
•	
•	
•	
AND	**DISPLAY** Abnormal plant behaviour display
THEN	Abnormal plant behaviour has been detected

6.3. The Knowledge Bases' Structures

The general organization of *Process Trainer* is given in figure 6.1. Each of the blocks indicated represents a different knowledge base that can be developed, tested, modified, used and even substituted, independently from the rest.

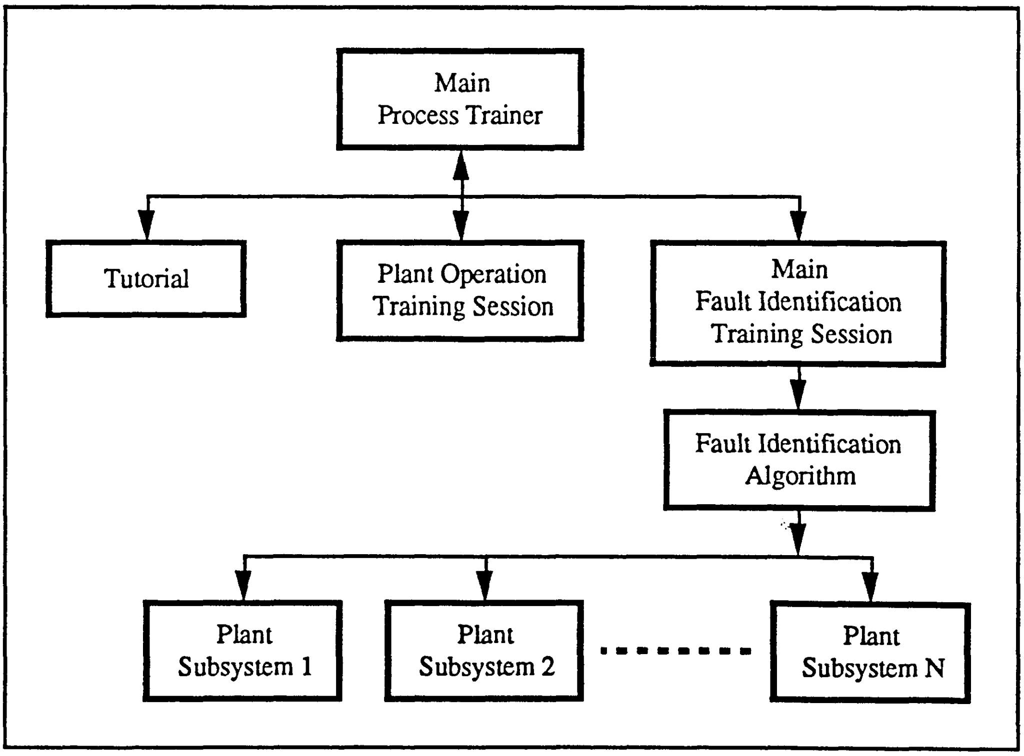


Figure 6.1. The knowledge base structure of *Process Trainer*

Furthermore, this arrangement offers three additional advantages: (i) each of the three training sessions can be used as a separate entity, (ii) different training programmes can be developed independently and then easily incorporated in *Process Trainer,* and (iii) the load on the computer processing individual knowledge bases is significantly reduced.

The *Main Process Trainer* knowledge base (table 6.15.) acts as the entry point to the CBT system. Its function is entirely organizational and does not include any instructional material. The first function it serves is to ask the student to register. It then creates an external file where a record of the subsequent training session(s) and the students performance will be kept.

Table 6.15. The data structure of the *Main Process Trainer* knowledge base

THE RULES OF THE *MAIN PROCESS TRAINER*
Group 1. Registration
Group 2. Selection of Training Programme
STRUCTURAL DESCRIPTIONS
Group 1. Relating to Registration
- TEXT
- EXPAND
- DISPLAY

Group 2. Relating to the Selection of a Training Programme
- TEXT
- EXPAND
- DISPLAY

Once that step is completed it directs the student to choose one of the three training sessions. Furthermore, it can provide explanatory information about each of the sessions upon a students request. Finally, based on the student's selection it will activate the corresponding instructional knowledge base.

A second knowledge base represents the *Tutorial* training programme that was described in section 5.2. In the *Tutorial* knowledge base (table 6.16.) rules and structural descriptions are used for the implementation of the tutor (Groups 1 to 4) and student models (Group 5) while domain specific descriptions are used to represent part of the domain model. In this way the functional structure of the knowledge base is independent of the domain. Thus, the adaptation of the *Tutorial* training programme to a new problem domain could only involve the substitution of the domain specific descriptions.

As previously described only part of the domain is represented in the actual knowledge base. The other part includes the *dynamic simulation problems* and the *operational sequences* which are stored outside *Process Trainer*. These are accessed via appropriate *communications rules* which are included in the groups describing the instructional material.

Similar to the *Tutorial* a single knowledge base describes the *Plant Operation Training Session* that was discussed in section 5.3. This knowledge base (table 6.17.) contains the tutor (Groups 1 to 3) and student models (Group 4) as rules and structural descriptions, and part of the domain model (domain specific descriptions).

Once again the structure of the training programme is independent of the domain and easily applicable to different problems. Finally, the use of the simulation based training system and of external graphics by the *Plant Operation Training Session* is also achieved from within the groups with the use of appropriate *communications rules*.

The *Fault Identification Training Session* (described in section 5.4.) should know or have the ability to identify the fault scenario that the student is studying. For this purpose a fault diagnosis algorithm needs to be implemented which will retain the ISA's flexibility for on-line instructor intervention. However, the fault diagnosis algorithm itself does not constitute part of the instructional process. Therefore, it is implemented as a separate knowledge base.

Table 6.16. The data structure of the *Tutorial* knowledge base

THE RULES OF THE *TUTORIAL*

Group 1. Selection of Training Session

Group 2. Introductory Session

- *Process*
- *Plant Structure*
- *Low Level Unit 1*
    - *Instructional Material for Low Level Unit 1*
        - Qualitative Description of Low Level Unit 1
- *Low Level Unit 2*...etc.
- *Medium Level Unit 1*
    - *Instructional Material for Medium Level Unit 1*
        - Qualitative Description of Medium Level Unit 1
        - Quantitative Description of Medium Level Unit 1
- *Medium Level Unit 2*...etc.

Group 3. Intermediate Session

- *Plant Subsystem1*
    - *Instructional Material for Plant Subsystem 1*
        - Qualitative Description of Plant Subsystem 1
        - Quantitative Description of Plant Subsystem 1
- *Plant Subsystem2*...etc.

Group 4. Advanced Session

- *Subsystems Interaction 1*
    - *Instructional Material for Subsystems Interaction 1*
        - Qualitative Description of Subsystems Interaction1
        - Quantitative Description of Subsystem Interaction1
- *Subsystems Interaction 2*...etc.

Group 5. Assessment of Student's Understanding

- *Pre-training Test*
- *Introductory Session Assessment*
    - *Assessment of Student's Understanding of Low Level Unit 1*
    - *Assessment of Student's Understanding of Low Level Unit 2*...etc.
    - *Assessment of Student's Understanding of Medium Level Unit 1*
    - *Assessment of Student's Understanding of Medium Level Unit 2*...etc.
    - *Assessment of Student's Understanding of Introductory Session*
- *Intermediate Session Assessment*
    - *Assessment of Student's Understanding of Plant Subsystem 1*
    - *Assessment of Student's Understanding of Plant Subsystem 2*...etc.
    - *Assessment of Student's Understanding of Intermediate Session*
- *Advanced Session Assessment*
    - *Assessment of Student's Understanding of Subsystem Interaction 1*
    - *Assessment of Student's Understanding of Subsystem Interaction 2*...etc.
    - *Assessment of Student's Understanding of Advanced Session*

DOMAIN SPECIFIC AND STRUCTURAL DESCRIPTIONS

Group 1.

- TEXT
- EXPAND
- DISPLAY

Group 2...etc.

The general structure of the *Fault Identification Training Session* was included in figure 6.1. It comprises a *Main Fault Identification Training Session* knowledge base, a *Fault Identification Algorithm* knowledge base, and a knowledge base for each of the *Problem Subsystems.*

Table 6.17. The *Plant Operation Training Session* knowledge base structure.

THE RULES OF THE *PLANT OPERATION TRAINING SESSION*

Group 1. Selection of Training Mode

Group 2. Demonstration Mode

Demonstration 1
Instructional Material for Demonstration 1
Qualitative Description of Demonstration Task 1
Dynamic Demonstration of Demonstration Task 1
Demonstration 2...etc.

Group 3. Interactive Mode

Interaction 1
Instructional Material for Interactive Study 1
Qualitative Description of Interactive Task 1
Interactive Execution Task 1
Interaction 2...etc.

Group 4. Assessment of Student's Understanding

Pre-training Test

Assessment of Demonstration Tasks
Assessment of Student's Understanding of Demonstration Task 1
Assessment of Student's Understanding of Demonstration Task 2...etc.

Assessment of Interactive Tasks
Assessment of Student's Execution of Task 1
Assessment of Student's Execution of Task 2...etc.

Assessment of Student's Understanding of Operational Sequences

DOMAIN SPECIFIC AND STRUCTURAL DESCRIPTIONS

Group 1.

TEXT
EXPAND
DISPLAY

Group 2...etc

The *Main Fault Identification Training Session* knowledge base (table 6.18.) serves three functions. Its first function is organizational. It allows the student to select the training mode as well as the fault scenario which will be studied. Also in this capacity, it accesses the simulation based training system and initiates the fault scenario, through the use of the appropriate *communications rules*. The second function of this knowledge base is to deliver the instruction under the demonstration mode. Since in a demonstration situation everything is known a priori it is not necessary to use the fault diagnosis algorithm. Finally, the last function of this knowledge base is to perform the evaluation of the student's understanding of the instructional material.

Table 6.18. The data structure of the *Main Fault Identification Training Session* knowledge base

THE RULES OF THE *MAIN FAULT IDENTIFICATION TRAINING SESSION*

Group 1. Selection of Training Mode

Group 2. Fault Diagnosis Demonstration Mode

- *Fault Scenario Demonstration 1*
    - *Instructional Material for Fault Scenario Demonstration 1*
        - Qualitative Description of Fault Scenario Demonstration 1
        - Dynamic Demonstration of Fault Scenario Demonstration 1
        - Demonstration of Diagnostic Strategy to Fault Scenario Demonstration 1
- *Fault Scenario Demonstration 2...etc.*

Group 3. Fault Diagnosis Interactive Mode

- *Fault Scenario Interaction 1*
    - *Dynamic Execution of Fault Scenario1*
- *Fault Scenario Interaction 2...etc*

Group 4. Assessment of Student's Fault Diagnosis Effectiveness

- *Pre-training Test*
- *Assessment of Student's Understanding of Fault Scenario Demonstration 1*
- *Assessment of Student's Understanding of Fault Scenario Demonstration 2...etc.*
- *End of Session Test*

DOMAIN SPECIFIC AND STRUCTURAL DESCRIPTIONS

Group 1.

- TEXT
- EXPAND
- DISPLAY

Group 2...etc.

The *Fault Identification Algorithm* knowledge base (table 6.19.) is activated when the student selects to study a fault scenario in the interactive mode. After the

Main Fault Identification Training Session knowledge base initializes the simulation based system for that particular scenario, the new knowledge base begins to monitor the process.

Table 6.19. The *Fault Identification Algorithm* knowledge base structure

THE RULES OF THE *FAULT IDENTIFICATION ALGORITHM*

Group 1. Receive Data Describing Plant Status
Group 2. Detect Abnormal Behaviour
Group 3. Group Affected Variables
Group 4. Identify Problem Subsystem
Group 5. Evaluate Operator's Response

When the fault scenario is activated (delayed *operational sequence)* the *Fault Identification Algorithm* knowledge base will detect the abnormal plant behaviour, diagnose the fault, and in due course, it will evaluate the effect of any student action. If it is unsuccessful or if there has been no student action then the *Fault Identification Algorithm* knowledge base will activate a new instructional knowledge base which describes the plant subsystem that includes the fault.

Finally, this last *Problem Subsystem* knowledge base (table 6.20.) will guide the student to identify the fault in the way described in section 5.4. (using the fault diagnosis strategy). Then when the student identifies the fault it will in turn re-activate the *Main Fault Identification Training Session* knowledge base so that a new fault scenario can be selected for study. Alternatively an assessment of the student understanding is carried out.

6.4. Summary

Chapter 5 described the training programmes that have been implemented in *Process Trainer*. In this chapter its functional definition in terms of its components was presented. *Process Trainer* is made up by a network of rule-based knowledge bases each serving a different function. Each knowledge base is constructed through the utilization of a set of general rules that have been defined.

Table 6.20. The data structure of a *Problem Subsystem* knowledge base

THE RULES OF THE *PROBLEM SUBSYSTEM*
Group 1. Identify Affected Variables
Group 2. Identify Problem Subsystem
Group 3. Identify Fault
Group 4. Second Problem Subsystem or Back to Beginning
DOMAIN SPECIFIC AND STRUCTURAL DESCRIPTIONS
Group 1.
    TEXT
    EXPAND
    DISPLAY
Group 2...etc.

This organization of the tutoring system offers several advantages. First, the definition of general rules permits the easy development of CBT systems that implement different training programmes. This was shown through the description of three different training programmes, each addressing a different training area. It is thus possible to use these rules as building blocks to develop different applications.

Second, the different ways in which the domain knowledge could be represented were examined. It was shown that part of the domain is represented within *Process Trainer,* while the other part is represented in each of the other modules of the ISA. Nevertheless, all domain knowledge is accessible at all times from any of the training programmes.

Third, it was shown that the domain specific knowledge is represented outside the main instructional structure of the knowledge bases. As a result the structure of the training programmes implemented can be retained and applied to different problem domains without great effort. Furthermore, modifications to the representations of the domain can be achieved without affecting the instructional process defined here.

Finally, it is pointed out that *Process Trainer* meets the broad functionalities termed as essential for flexible, easy to use and easy to upgrade CBT systems that were identified in section 3.6. In the next chapter the adaptation of *Process Trainer* and its application to a particular case study is examined.

Chapter 7

PROCESS TRAINER:
AN ILLUSTRATIVE APPLICATION

The two previous chapters described in detail the training programmes, the structure and the tools that are offered in *Process Trainer* which enable its application to different problem domains. In this chapter an application of *Process Trainer* and the ISA to a specific problem domain is presented. At this point it should be stressed that other than the specific domain descriptions, everything else is of general applicability.

The simple illustrative application aims to demonstrate the functionalities of *Process Trainer* and the ISA (rather than be a complete training course). It has been developed and implemented around an evaporator pilot plant which is one of three pilot plants in the Department of Chemical Engineering at Imperial College. The evaporator is regularly used for an operation and analysis project by all first year undergraduates.

It will, thus, be shown that it is possible to implement 'comprehensive' training programmes (which address a range of training needs at different levels of detail and complexity), through a stand-alone computer based training system which is flexible, easy to develop and easy to support and update.

7.1. Plant Description

A schematic diagram of the process is given in figure 7.1. Its purpose is to concentrate an aqueous solution of potassium nitrate in a forced recirculation loop at a rate of up to 30 grams per second. The solution circulates via a centrifugal pump at one to three kilograms per second through the tubes of a heat exchanger (or calandria) were heat is supplied by steam introduced in the shell side.

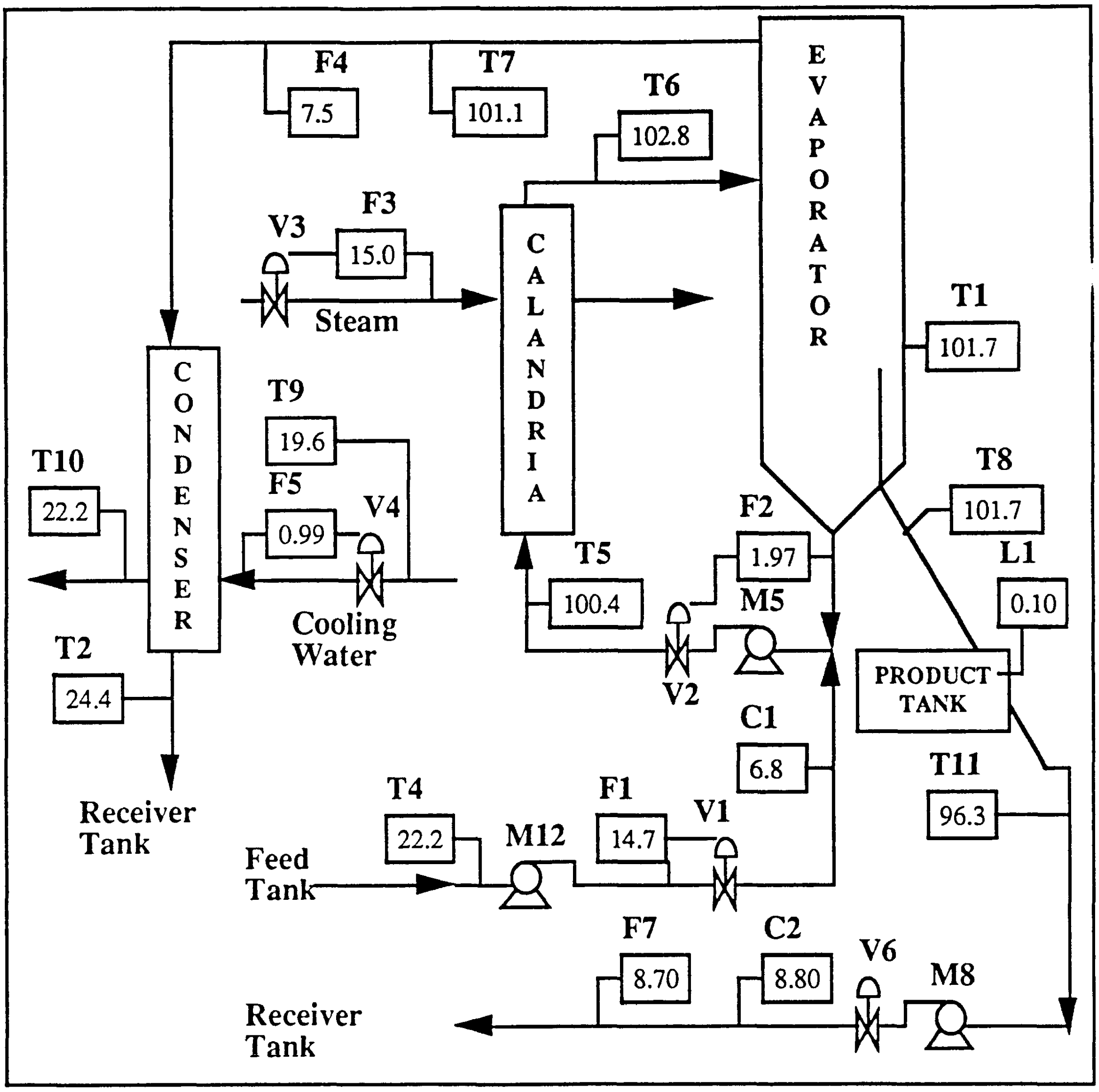


Figure 7.1. Schematic diagram of evaporator pilot plant

The hot, one or two phase fluid then passes to a vapour-liquid separator (splitter) where the phases disengage and the liquid returns to the recirculation pump by gravity. Also by gravity, the concentrated product solution overflows through a stand pipe in the splitter to the receiver tank from where it is pumped to a large vessel. The vapour from the splitter passes through the shell of a condenser with cooling water on the tube side, and flows by gravity as a liquid into the same large vessel.

The process is monitored, operated and controlled through monitors in a control room under RTPMS.

7.2. Task Analysis and Educational Objectives

First year students are asked to perform tasks similar to the ones that process operators perform in an industrial setting. In particular students are asked to:

1. Decide upon a set of start-up and shut-down procedures.
2. Implement the pre-specified control configuration.
3. Implement these procedures successfully.
4. Achieve and maintain a steady state by manipulating the controls.
5. Obtain a desired product quality and quantity.

Students first become familiar with the instrumentation and the control of the process, as well as the location and function of all plant items. They determine the location of all control and manual valves and meters, and all safety valves. Next they need to understand the process, how meters operate, how control valves work, how the links are made via the computer, and finally learn to use the control system.

Currently, the students discuss all relevant issues with a tutor and a demonstrator on location. The development of operating procedures usually follows a hierarchical task analysis which includes identification of major subsystems (cooling, recirculation loop, feed system, etc.) and constraints (i.e. is it better to get the steam on before cooling, etc.). Furthermore, they engage in self-study that helps them understand certain issues.

Familiarisation with the control system has a hands-on nature. All control system functions can take place without the plant actually being operated. Therefore, the students get the chance to practice closing cascades and performing various control actions without them affecting the operation. The inter-relationships within the plant and cause and effect relationships are learned on-the-job.

The nature of the present instructional approach and the experiences associated with it indicate that the training exercise can greatly be enhanced by addressing a number of shortcomings in the present practice. First, students could benefit from more adequate preparation prior to being exposed to the actual plant. Second, smaller groups of students and thus more active participation by all will have more desirable effects. Third, experience with faults and emergencies prior to actual plant operation will make the real-plant experience more time-efficient and effective. Fourth, facilities to decide on the best out of several alternative operational procedures could be provided so that students can experience the effects of major choices. These issues are addressed by *Process Trainer* and the ISA.

7.3. The Simulation Based Training System

A simulation based training system (SBTS) following the specifications set out in section 4.2.2. was initially implemented for the evaporator pilot plant. This can be used independently (Kassianides and Macchietto, 1990) or can be accessed to provide *Process Trainer* with dynamic simulation capabilities. Here the SBTS is examined in view of its use by *Process Trainer*.

7.3.1. Dynamic Simulation Problems

In the description of the training programmes employed by *Process Trainer* the need was identified for several dynamic simulation problems. The definition of a simulation problem includes the simulation models, the initial conditions, and the control system functions (configuration and initial conditions) that correspond to the simulation model. However, the foundation of a simulation problem is the actual simulation model. Once this has been implemented the other two components can easily be defined.

More specifically models were used which describe: (i) individual plant subsystems, (ii) the interactions of plant subsystems, and (iii) the whole plant. In the application described the development of the dynamic simulation problems was based on a simulation model of the whole plant.

Complete Evaporator Pilot Plant Simulation Model

The first attempt to develop a dynamic simulation for the evaporator pilot plant was carried out by Bernal (1986) using the SpeedUp dynamic simulator (section 4.1.2.). Bernal's initial model was then revised by Rozo (1987) who showed that the simulation accurately predicted plant behaviour for a wide range of operating conditions. The model is based on differential and algebraic equations, with rigourous physical properties and parameters fitted to plant data. Rozo also modified the dynamic model so that it could be run in real-time in conjunction with RTPMS.

The resulting model offered a close representation of the real system with similar dynamics, and a description of most unit operations in the process. Manual valves, power switches, and pumps were modeled as zero-one variables and were mapped onto corresponding RTPMS switches which enabled the operator to alter their status by turning the respective switches either ON or OFF. On the contrary proper dynamic models were developed for the main unit operations in the plant (feed tank, calandria, splitter, condenser, and product tank). These were developed as individual SpeedUp models which included the low level components mentioned above and as such represent the five main plant subsystems.

It was thus possible to manually execute all procedures necessary for start-up, shut-down, normal operation and fault handling via the RTPMS consoles. The model also catered for situations where draining of the system could be performed. However, there were a number of problems with Rozo's simulation.

The most important problem was that the simulation model was structured and written in such a way that led to a considerable amount of discontinuities which affected its effectiveness. Most significantly, the level of discontinuities at the onset of boiling was such that the numerical integration had difficulty to converge, and was not always able to continue past the boiling stage. Additionally, even if transition past the boiling point was achieved the accumulated effect of the various discontinuities on processing efficiency resulted in the simulation lagging so much with respect to real time that it was unsuitable for use as a training tool.

Problems were also identified in the condenser model of the simulation. The outlet temperature of the product stream of the condenser was set to a specific temperature. As a result cutting off the cooling water supply when boiling had been achieved (for example by closing a manual valve) led to the simulation crashing because of a floating point error in the simulation. Other areas where the simulation could be improved and thus increase the fidelity of the simulation were identified and reported by Rozo (1987). However, these are not of integral importance to the use of the simulation for training purposes and therefore, they have not been examined.

After careful study of the model the sources of most discontinuities were identified and modified. Minor modifications to the definition of operational constraints were carried out for each model. All operational constraints were removed from the individual model definitions and were defined as set variables. Finally, values representative of an initial solution were pre-set to enable the easy initialisation of the simulation.

A modification of the condition controlling boiling in the simulation resulted in the system achieving the boiling state without any problems and thus the simulation's real time performance was improved without affecting the representation of the overall plant dynamic behaviour. Furthermore, in order to increase the processing efficiency of the simulation model the conditions for draining were removed since the objectives as defined by the use of the simulation as a training tool did not require their presence. As a result the evaporator pilot plant simulation now runs in real time in a multiuser environment even without any special treatment with respect to CPU allocation.

Another modification to Rozo's model was made in order to take advantage of SpeedUp's External Data Interface (EDI) program and the database server (section 4.3.) that made the communication with RTPMS more efficient. The facilities offered and utilized through this communication link were described in detail in section 4.1.2.

Further modifications were made to the condenser model in order to address the problem mentioned above. These involved the inclusion of appropriate conditional statements which control the execution of a model and avoid the case which leads to the floating point error while giving a representative value for the outlet temperature. Finally, the last modification to the model was the inclusion of the sample error model described in section 4.1.2. to the feed flowrate. Table 7.1. gives some statistics on the model size and also provides a summary of all the plant features that have been incorporated in the Evaporator Pilot Plant Model.

Plant Subsystem Models

As described above the main unit operations were already described as individual models which also incorporate the low level units. As such the broad definition of the plant subsystems has already been made. In order to obtain individual plant subsystem models one has to extract them from the model of the whole plant along with any related information (for example, initial conditions, pre-set values, etc.) and create a new independent simulation model. Thereafter, it is necessary to define any new inputs which would have been evaluated by subsystems upstream in the complete plant structure and map these to respective RTPMS variables.

One such exercise has been carried out for the condenser subsystem. The selection of the condenser subsystem was based on the original need to modify it in view of the complete plant model. Therefore, the condenser subsystem model was extracted and modified in isolation. Then it was again introduced in the complete plant model. However, the stand alone condenser subsystem model was retained for use by *Process Trainer* in the intermediate session of the *Tutorial.*

Table 7.1. Plant features represented in the Evaporator Pilot Plant Model

Model Statistics

Total System Size (knowns and unknowns)	**249**
Number of Variables/Equations	**169**
Number of Unknown Derivatives	**27**
Number of Set Variables	**53**
Number of Initial Variables	**27**
Number of Pre-set Variables	**24**

Unit Operations Represented as ON - OFF switches

1. Five Manual Valves on the STEAM line of the Heat Exchanger
2. Three Manual Valves on the Cooling Water Supply to the Condenser
3. One Manual Valve on the Feed line
4. Two Main Power Switches
5. One Air Dryer
6. One Agitator
7. A Recirculation Pump
8. A Transfer Pump
9. A Feed Pump

Unit Operations Represented by a Dynamic Model

1. A Feed Tank
2. A Calandria (Heat Exchanger)
3. A Splitter
4. A Product Tank (Buffer tank)
5. A Condenser
6. The Steam Control Valve
7. The Feed Control Valve
8. The Recirculation Control Valve
9. The Liquid Product Control Valve
10. The Cooling Water Control Valve

Error Models

1. Feed Flowrate

Models describing Plant Subsystem Interactions

Similar to the case of individual plant subsystems, models which describe the interactions between plant subsystems can easily be extracted from the complete plant simulation model. In this case the exercise would involve the extraction of two or more models from the complete plant definition and the definition of a new independent plant model. Once again the appropriate mapping to RTPMS has to be performed.

Initial Conditions

The incorporation and use of appropriate initial conditions for the initialisation of a simulation model was described above. This approach was followed for the condenser subsystem model which is initialized around a steady state. Nevertheless, if the same simulation model is to be used in a range of operational situations then corresponding initial conditions need to be defined. There is no limit on the possible initialisation points for a simulation model. However, in view of the use of the complete plant simulation model within *Process Trainer* three specific sets of initial conditions were defined.

The first set of initial conditions corresponds to a pre-start-up stage and thus permits the use of the simulation for plant start-up studies. These conditions describe the plant in its cold state with no flows, all temperatures at room level, all manual valves closed, pumps shut, etc.

The second and third sets of initial condition initialize the dynamic simulation at a point were boiling has been achieved and the plant is operating in steady state. However, each describes a situation with different heating rates, achieved by different steam supply levels. One is representative of a situation characteristic of slow boiling while the other describes fast boiling. The second and third sets permit the study of cause and effect relationships around a steady state, as well as provide representative starting points for plant shut-down studies.

Control System Functions

Chapter 4 described in detail the functionalities and the use of the control system module as part of the ISA. In this particular application the control interfaces used for training are an exact replica of those on the plant. All the measurements that are available from the real plant have been created as variables that are received from the simulation. The use of RTPMS switches that map onto zero-one variables in the simulation was mentioned above. Additionally, all five control loops were implemented in exactly the same way as they are implemented for controlling the actual plant. These were then tuned according to the simulation so that they demonstrate realistic behaviour. Finally, a range of displays have been created. These include mimic diagrams of the process, graphs showing the dynamic profiles of variables, displays showing the status of unit operations, etc.

7.3.2. Operational Sequences

The use of operational sequences in the instructional process was described in detail in chapter 5 while the advantages and use of the operations management module as part of the SBTS was described in detail in chapter 4. In summary the definition of operational procedures in SUPERBATCH involves two steps. The first step is the definition of control primitives (phases) which define a particular action that will be executed. Then in a second step these control primitives are assembled to define formal operational procedures (master procedures) that can be scheduled for execution.

Control Primitives (Phases)

The definition of a control primitive in itself involves two steps. The first step is the definition of the phase in RTPMS. This involves the creation of a set of RTPMS tags and the implementation in RTPMS of algorithms that will execute the desired actions (for example TURN ON RTPMS switch 1, put a control valve on automatic then close cascade etc.). In essence these phases can be activated and used directly from RTPMS's interface. Once these phases are defined in RTPMS then they are described in SUPERBATCH which will control their automatic execution.

In the evaporator pilot plant application phases have been defined to carry out a variety of tasks, from managing the execution of the simulation to performing actual operational actions. The set of phases that have been developed are listed in table 7.2.

Operational Procedures (Master Procedures)

Once the phases have been developed then one can easily define master procedures that will execute these phases in a predetermined order and store them in operational sequences models. Two different approaches have been followed for the definition of the master procedures.

Table 7.2. Control primitives for the evaporator pilot plant

Phases Describing Operational Actions (Total number: 47)

1. Two phases for each of the Manual Valves. One phase will OPEN the Manual Valve and another will CLOSE it (Phases total: 18).
2. Two phases for each of the Power Switches. One phase will turn them ON and another will turn them OFF (Phases total: 4).
3. Two phases for the Air Dryer. One phase will turn it ON and another will turn it OFF (Phases total: 2).
4. Two phases for the Agitator. One phase will turn it ON and another will turn it OFF (Phases total: 2).
5. Two phases for each of the Pumps. One phase will turn them ON and another will turn them OFF (Phases total: 6).
6. Three phases for each of the Control Loops. One phase will set up a control loop, another will set it up and put a setpoint and a third will take it off (Phases total: 15).

Phases For Control of the SBTS (Total number: 5)

7. Two phases for introducing a time delay in the execution of a schedule. One phase allows the prior definition of the time delay and another the allows its interactive definition (Phases total: 2).
8. One phase to start the simulation (Phases total: 1).
9. One phase to pause the simulation (Phases total: 1).
10. One phase to abort the simulation (Phases total: 1).

In the first approach each individual phase has been described as a very simple master procedure that can be scheduled independently. This detailed definition of each operational action as an operational sequence has two applications. The first application is during the interactive mode of the *Plant Operation Training Session* of *Process Trainer*. Here the student is allowed to interactively define his own sequence of actions in order to achieve the operational objectives. The second application of this approach is in the *Fault Identification Training Session* of *Process Trainer*. Here the tutor can interactively schedule individual or a multiple plant upsets. For example a tutor may interactively schedule the closure of a manual valve and also define when this action should take place.

The second approach involves the more traditional use of SUPERBATCH. Here a group of phases is assembled to define a more substantial operational sequence such as a complete start-up procedure. The main application of this approach is in the execution of demonstrations relating to plant operation These may be used to demonstrate the effect of control actions on the process dynamics, operational sequences, etc. Another application of this approach is the formal definition of fault scenarios that are used by the *Fault Identification Training Session* of *Process Trainer.*

The possible combinations of these phases in master procedures and then the combination of master procedures in operational sequences models is infinite and many different combinations have been used in the evaporator pilot plant application. A representative list of operational sequences models that have been developed for the evaporator application is given in table 7.3. while table 7.4. lists representative master procedures that have been developed.

Table 7.3. Operational Sequences Models for the evaporator pilot plant

1. Interactive Start-Up with 1 Operator
2. Demonstration of Plant Start-Up with 1 Operator
3. Demonstration of Plant Start-Up with 2 Operators...etc.
4. Demonstration of Plant Shut-Down with 1 Operator
5. Demonstration of Plant Shut-Down with 2 Operators...etc.
6. Demonstrations of Operational Sequences Group 1
7. Demonstrations of Operational Sequences Group 2...etc.
8. Fault Scenarios Group 1
9. Fault Scenarios Group 2...etc.

Table 7.4. Master Procedures (MP) for the evaporator pilot plant

1. All phases described in table 7.2. have been described as MPs
2. MP for Setting up the Liquid Circulation in the Plant
3. MP for Starting the Feed Supply with 1 operator (another for 2 operators up to 5)
4. MP for Starting the Cooling Water Supply (2 operators...etc)
5. MP for Starting the Steam Supply (2 operators...etc)
6. MP for Turning ON the Power and the Auxiliary Equipment (2 operators...etc.)
7. MP for Stopping the Feed Supply (2 operators...etc.)
8. MP for Stopping the Cooling Water Supply (2 operators...etc.)
9. MP for Stopping the Steam Supply (2 operators...etc)
10. MP for Turning OFF the Power and the Auxiliary Equipment (2 operators...etc.)
11. MP for plant Start-Up (2 operators...etc.)
12. MP for plant Shut-Down (2 operators...etc.)
13. MP for Demonstration Scenario 1 (Dynamic response of the Feed Control Loop)
14. MP for Demonstration Scenario 2...etc.
15. MP for Fault Scenario 1 (Disable the Steam Control Valve)
16. MP for Fault Scenario 2...etc.

Control System Functions

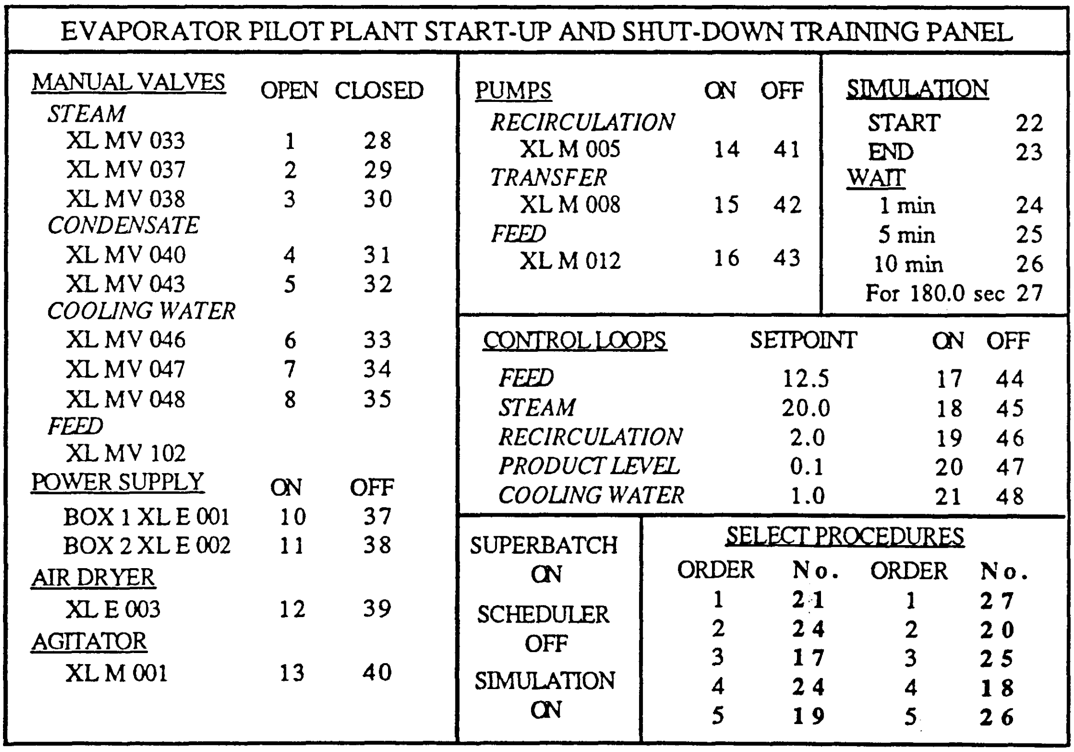
The use of the control system functions for the implementation of the phases was described above. Another application of the control system arises from the fact that it acts as the front end to SUPERBATCH. The actual scheduling of the master procedures which are included in the operational sequences model is performed at the control system level through appropriate displays that map onto the operational sequences models.

Figure 7.2. represents one RTPMS display that is available to students for the interactive definition of start-up or shut-down procedures. The major part of the display lists all actions that an operator can choose along with an associated number describing that action. For example the procedure for opening the XL MV 033 steam manual valve is **1**, while that for switching off the recirculation pump XL M 005 is **41**. Furthermore, the student can choose to set up the various control loops and

specify the desired setpoint. Other features are initiating or ending the simulation, and specifying a wait time before the next action is executed.

At the bottom right corner of the display there is a point of entry for up to ten actions at a time. It is here that the operator inputs the corresponding numbers describing the desired sequence of events. Once the operator enters a sequence of procedures (one to ten) then he can schedule them by turning the scheduler switch on. Following that he can then schedule another ten or less and so on.

Finally he can sit back and evaluate his choice according to its effectiveness in achieving the desired targets. The order of the numbers shown in the Select Procedures section in figure 7.2. corresponds to a specific set of actions to be executed. These are described in table 7.5.



EVAPORATOR PILOT PLANT START-UP AND SHUT-DOWN TRAINING PANEL

MANUAL VALVES	OPEN	CLOSED
STEAM		
XL MV 033	1	28
XL MV 037	2	29
XL MV 038	3	30
CONDENSATE		
XL MV 040	4	31
XL MV 043	5	32
COOLING WATER		
XL MV 046	6	33
XL MV 047	7	34
XL MV 048	8	35
FEED		
XL MV 102		
POWER SUPPLY	ON	OFF
BOX 1 XL E 001	10	37
BOX 2 XL E 002	11	38
AIR DRYER		
XL E 003	12	39
AGITATOR		
XL M 001	13	40

PUMPS	ON	OFF
RECIRCULATION		
XL M 005	14	41
TRANSFER		
XL M 008	15	42
FEED		
XL M 012	16	43

SIMULATION	
START	22
END	23
WAIT	
1 min	24
5 min	25
10 min	26
For 180.0 sec	27

CONTROL LOOPS	SETPOINT	ON	OFF
FEED	12.5	17	44
STEAM	20.0	18	45
RECIRCULATION	2.0	19	46
PRODUCT LEVEL	0.1	20	47
COOLING WATER	1.0	21	48

SUPERBATCH
ON
SCHEDULER
OFF
SIMULATION
ON

SELECT PROCEDURES

ORDER	No.	ORDER	No.
1	21	1	27
2	24	2	20
3	17	3	25
4	24	4	18
5	19	5	26

Figure 7.2. Panel for Interactive Definition of Start-Up and Shut-Down Procedures for the Evaporator Pilot Plant

Table 7.5. The specific actions scheduled in figure 7.2.

ORDER	PROCEDURE	DESCRIPTION
1	21	Set Up the Cooling Water Control Loop
2	24	Wait For One Minute
3	17	Set Up the Feed Control Loop
4	24	Wait For One Minute
5	19	Set Up the Recirculation Control Loop
6	27	Wait For Three Minutes
7	20	Set Up the Product Level Control Loop
8	25	Wait For Five Minutes
9	18	Set Up the Steam Control Loop
10	26	Wait For Ten Minutes

7.4. The Adaptation of *Process Trainer*

Chapter 6 examined in detail the features, characteristics, and knowledge base structure of *Process Trainer* and its individual components. Then section 7.3. examined the application of the SBTS for the representation of that part of the Evaporator Pilot Plant domain which as described in section 6.2. is not represented within *Process Trainer*. In this section the actual adaptation of *Process Trainer* to the Evaporator Pilot Plant case study is described through the examination of its training programmes (figure 7.3.).

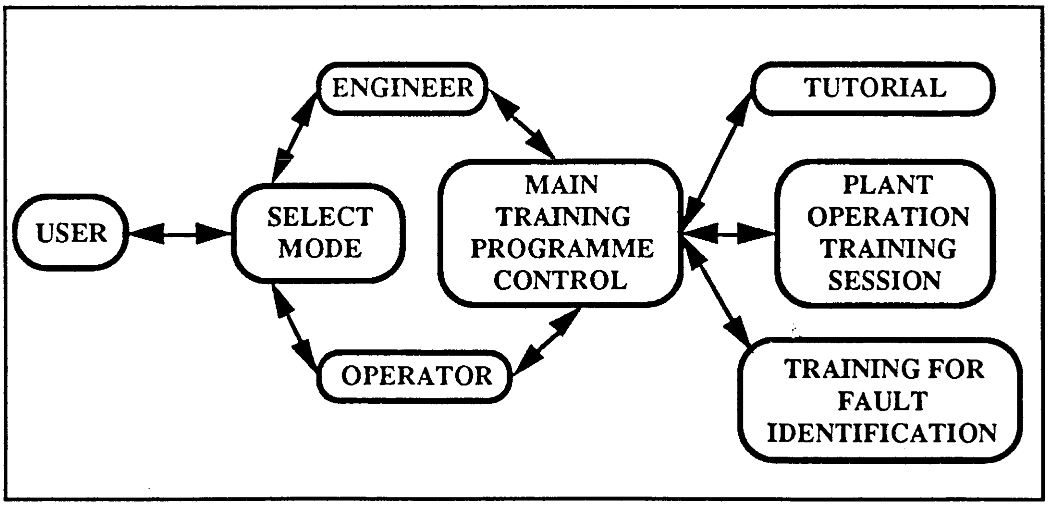


Figure 7.3. The overall structure of *Process Trainer*

7.4.1. The Evaporator Pilot Plant *Tutorial*

General descriptions of the instructional process, the instructional material and the general knowledge base structure of the *Tutorial* were provided in the two previous chapters. Therefore, the description of the *Tutorial* application in this section will only consider the aspects which relate to the particular problem domain of the Evaporator Pilot Plant.

The *Tutorial* is based on definition of plant subsystems and their respective low and medium level units. In the evaporator pilot plant five main plant subsystems have been identified. These are presented in tables 7.6. with their corresponding plant components (low and medium level units) and their associated RTPMS measurements.

Table 7.6.a. The Heat Exchanger Subsystem of the Evaporator Pilot Plant

Heat Exchanger Subsystem

Low Level Units

1.	Power Supply Box 1	XL E 001
2.	Power Supply Box 2	XL E 001
3.	Air Dryer	XL E 003
4.	Manual Valve	XL MV 033
5.	Manual Valve	XL MV 037
6.	Manual Valve	XL MV 038
7.	Manual Valve	XL MV 040
8.	Manual Valve	XL MV 043
9.	Control Valve	XL V 002
10.	Control Valve	XL V 003
11.	Concentration Meter	XL C 001
12.	Thermometer	XL T 005
13.	Thermometer	XL T 006
14.	Flowmeter	XL F 001
15.	Flowmeter	XL F 002
16.	Flowmeter	XL F 003
17.	Recirculation Pump	XL M 005

Medium Level Units

1. Heat Exchanger
2. Control Loop between XL F 001 and XL V 001
3. Control Loop between XL F 003 and XL V 003

Table 7.6.b. The Feed Subsystem of the Evaporator Pilot Plant

Feed Subsystem

Low Level Units

1.	Power Supply Box 1	XL E 001
2.	Power Supply Box 2	XL E 001
3.	Air Dryer	XL E 003
4.	Agitator	XL M 001
5.	Manual Valve	XL MV 102
6.	Control Valve	XL V 001
7.	Concentration Meter	XL C 001
8.	Thermometer	XL T 004
9.	Flowmeter	XL F 001
10.	Feed Pump	XL M 012

Medium Level Units

1. Feed Tank
2. Control Loop between XL F 001 and XL V 001

Table 7.6.c. The Evaporator Subsystem of the Evaporator Pilot Plant

Evaporator Subsystem

Low Level Units

1.	Power Supply Box 1	XL E 001
2.	Power Supply Box 2	XL E 001
3.	Air Dryer	XL E 003
4.	Thermometer	XL T 001
5.	Thermometer	XL T 006
6.	Thermometer	XL T 007
7.	Thermometer	XL T 008
8.	Flowmeter	XL F 004

Medium Level Units

1. Splitter

Introductory or Plant Components Session

The description of the five subsystems indicates the various low and medium level units that the student needs to understand before he studies the more complicated aspects of plant operation. Furthermore, it was previously stated (section 5.1.) that the main objective of this session is to convey the student a broad understanding of the various plant components and how they contribute or affect plant operation. This constraint permits emphasis to be placed on the functionality of each component category. Therefore, where possible, a representative type of each component category was examined and variations amongst the actual component types that are in place on the real plant were ignored.

Table 7.6.d. The Condenser Subsystem of the Evaporator Pilot Plant

Condenser Subsystem

Low Level Units

1.	Power Supply Box 1	XL E 001
2.	Power Supply Box 2	XL E 001
3.	Air Dryer	XL E 003
4.	Manual Valve	XL MV 046
5.	Manual Valve	XL MV 047
6.	Manual Valve	XL MV 048
7.	Thermometer	XL T 002
8.	Thermometer	XL T 007
9.	Thermometer	XL T 009
10.	Thermometer	XL T 010
11.	Flowmeter	XL F 004
12.	Flowmeter	XL F 005
13.	Control Valve	XL V 005

Medium Level Units

1. Condenser
2. Control Loop between XL F 005 and XL V 005

Table 7.6.e. The Product Tank Subsystem of the Evaporator Pilot Plant

Product Tank Subsystem

Medium Level Units

1.	Power Supply Box 1	XL E 001
2.	Power Supply Box 2	XL E 001
3.	Air Dryer	XL E 003
4.	Control Valve	XL V 006
5.	Concentration Meter	XL C 002
6.	Thermometer	XL T 008
7.	Thermometer	XL T 011
8.	Flowmeter	XL F 007
9.	Level Meter	XL L 001

Medium Level Units

1. Condenser
2. Control Loop between XL L 001 and XL V 006

Figure 7.4. shows the overall structure of the introductory session of the evaporator pilot plant *Tutorial* and also the extent of development that has currently been performed for this illustrative application. The instructional process as well as

representative displays used for the presentation of the instructional material were given in section 5.1.

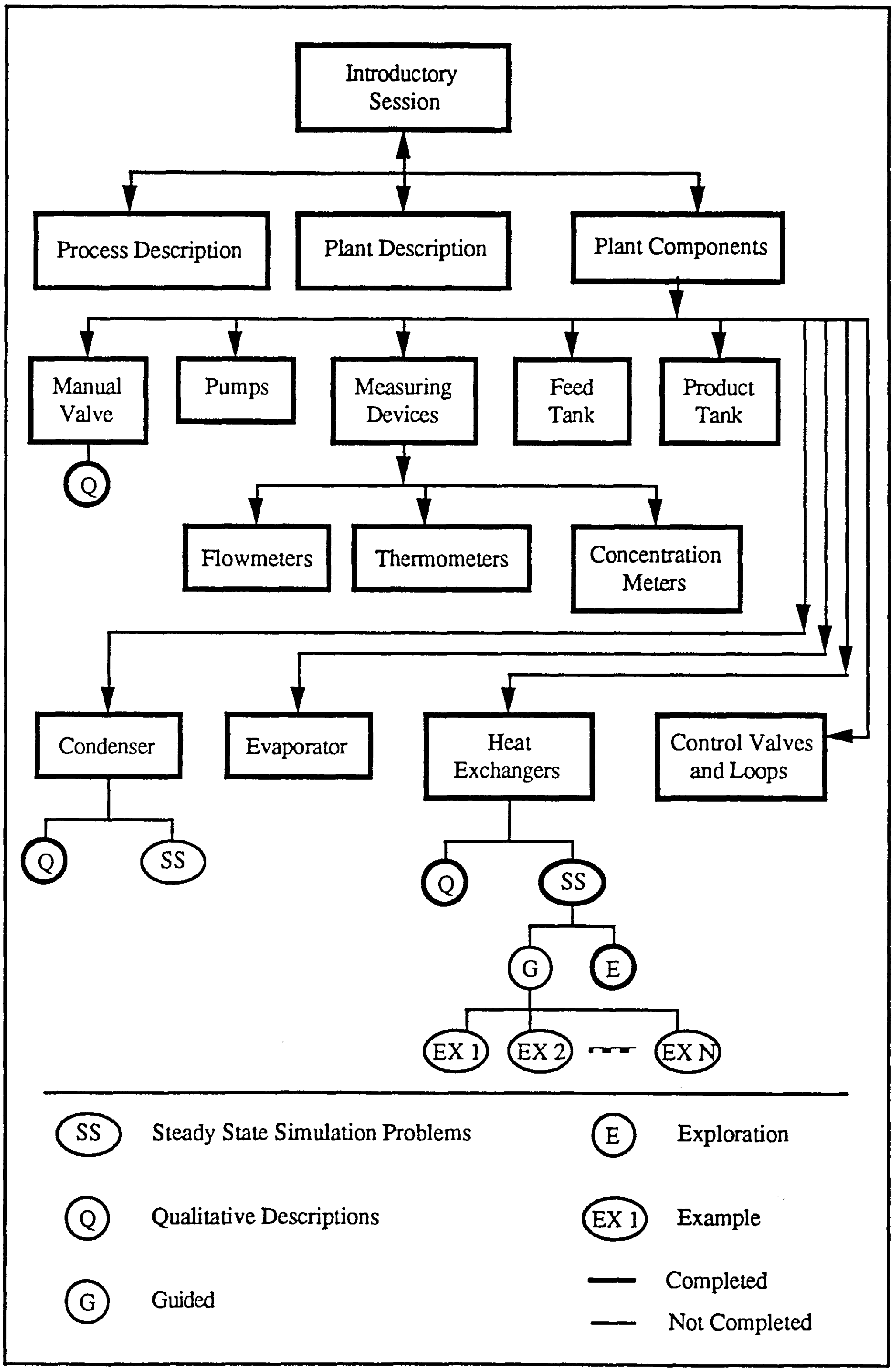


Figure 7.4. The Introductory Session of the Evaporator Pilot Plant *Tutorial*

Intermediate or Plant Subsystems Session

The five subsystems and their plant components were described in tables 7.6. A detailed description of the instructional process used to convey an understanding of the plant subsystems to students, as well as examples of student interactions were presented in section 5.1. The general structure and facilities offered by the intermediate session for the evaporator pilot plant as well as the extent of development for the illustrative application are shown in figure 7.5.

Advanced or Plant Subsystems Interactions Session

As with the two previous sessions the instructional process used to convey the understanding of the interactions between the plant subsystems to students as well as examples of interactions were also presented in section 5.1. The interactions which are studied in the evaporator pilot plant application are given in table 7.7.

Table 7.7. Interactions between the Subsystems of the Evaporator Pilot Plant

Interactions Studied

1. Feed and Heat Exchanger Subsystem Interactions
2. Heat Exchanger and Evaporator Subsystems Interactions
3. Feed, Heat Exchanger and Evaporator Subsystems
4. Feed, Heat Exchanger, Evaporator, and Condenser Subsystems
5. The Complete Plant

First the interaction of the feed and heat exchanger subsystems is studied. Then the interaction between the heat exchanger subsystem and the evaporator subsystem. This is a deviation of the formal approach of adding a new downstream unit to the units whose interaction has been studied. However, it is necessary because the heat exchanger and evaporator subsystems are closely interdependent due to the recycle flow. Third the interactions between all three units are examined. Fourth, the condenser unit is added to the set of subsystems whose interaction has already been described. Finally the interactions within the complete plant structure are presented.

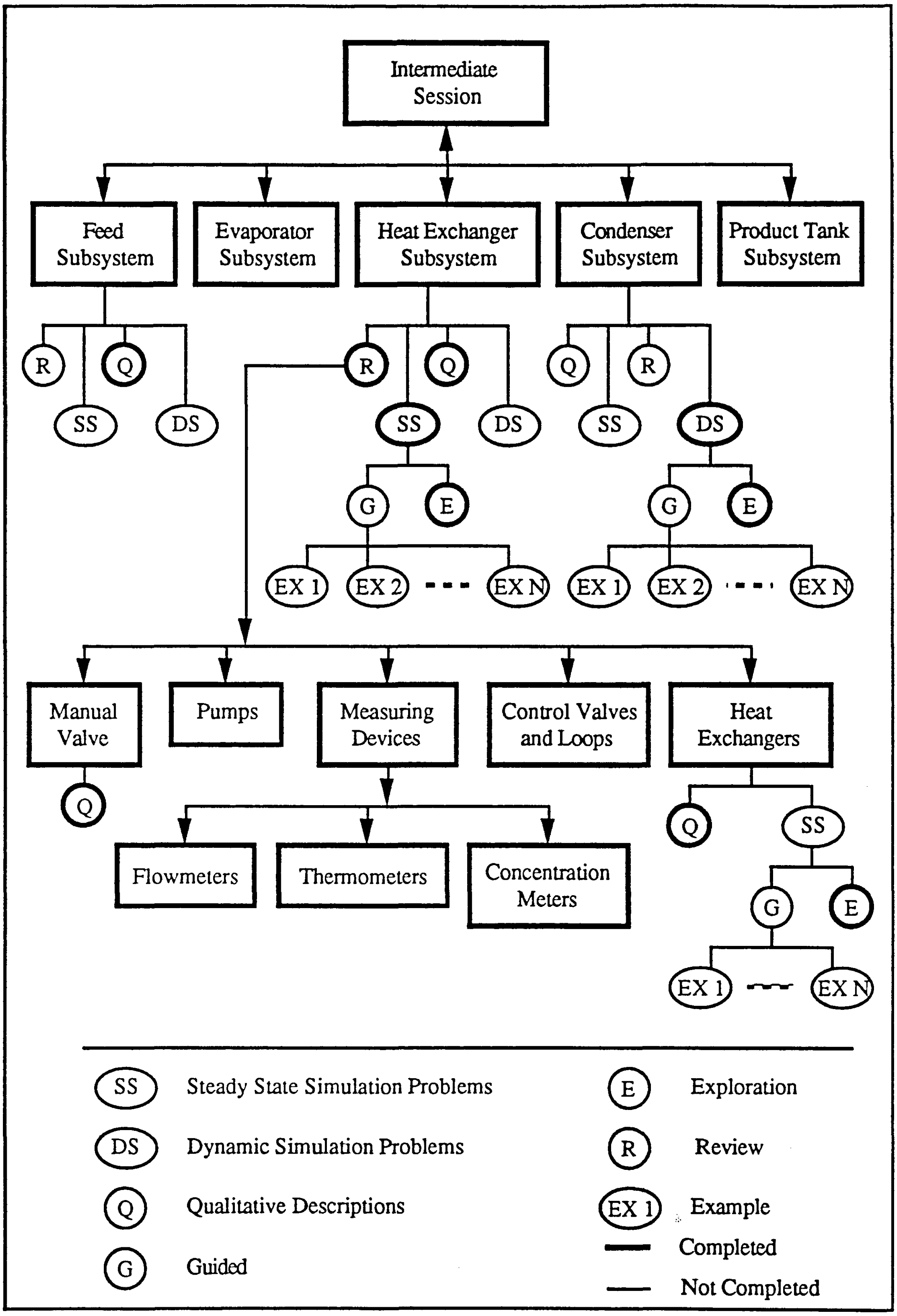


Figure 7.5. The Intermediate Session of the Evaporator Pilot Plant *Tutorial*

Last the general structure and facilities offered by the advanced session for the evaporator pilot plant are given in figure 7.6. where the extent of development for the illustrative application is also shown.

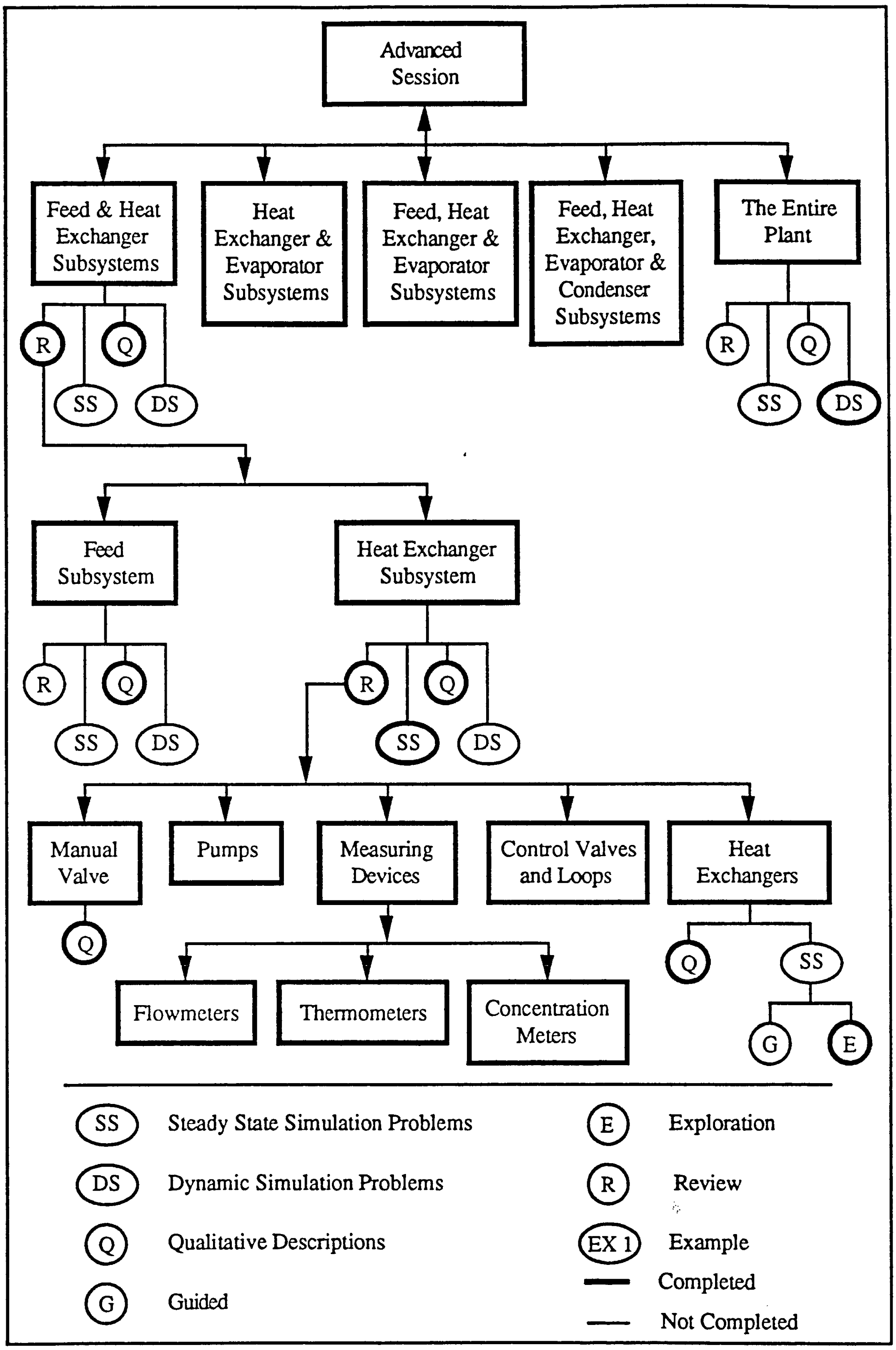


Figure 7.6. The Advanced Session of the Evaporator Pilot Plant *Tutorial*

7.4.2. The Evaporator Pilot *Plant Operation Training Session*

Similar to the evaporator pilot plant *Tutorial,* detailed descriptions of the instructional process, the instructional material and the general knowledge base structure of the Plant Operation Training Session were provided in chapters 5 and 6 of this thesis. As mentioned the foundation of this training programme is based on the availability of a SBTS. The SBTS for the evaporator pilot plant was described in detail in section 7.3. where also representative demonstrations and operational sequences were presented. The general structure of the Plant Operation Training Session for the evaporator pilot plant is given in figure 7.7.

7.4.3. The Evaporator Pilot Plant *Fault Identification Training Session*

Finally, this section examines the *Fault Identification Training Session* for the evaporator pilot plant. The functionalities of general structure of this session are the same as those offered by the structure of the *Plant Operation Training Session* for the evaporator pilot plant until the point where fault detection is required. A functional definition of what the rest of the structure entails from this point onwards is given in figure 7.8.

Two major areas can be identified. The first is the system detection of abnormal plant behaviour and the diagnosis of faults. It was previously mentioned that this was performed by the *fault identification algorithm* knowledge base through the appropriate use of *heuristics.* Finally, the second area is the actual dialogue between the system and the student which aims to convey to the student the fault diagnosis strategy described in section 5.3. A representative dialogue was also presented in the same section (figure 5.10).

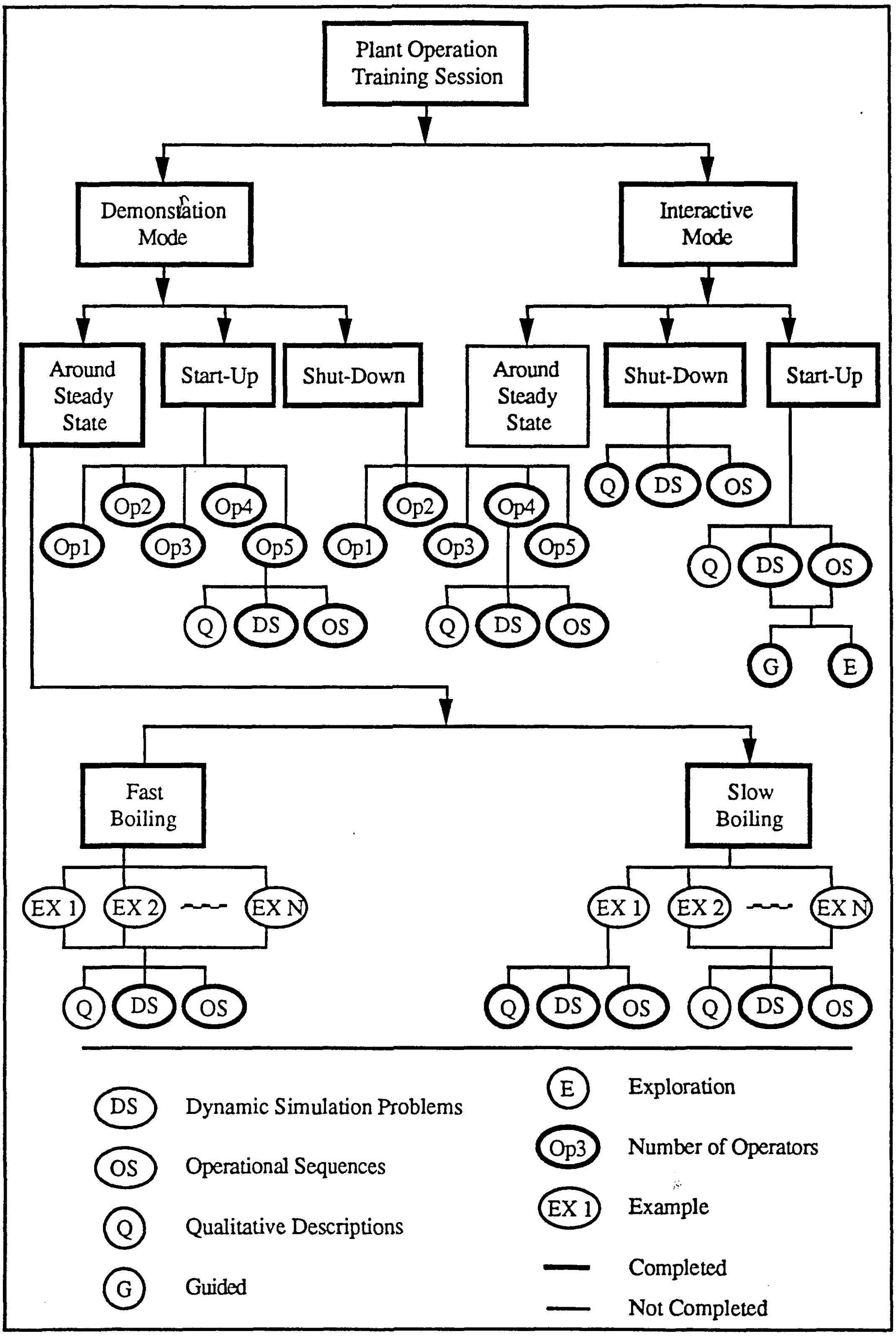


Figure 7.7. The overall structure of the Evaporator Plant Operation Training Session

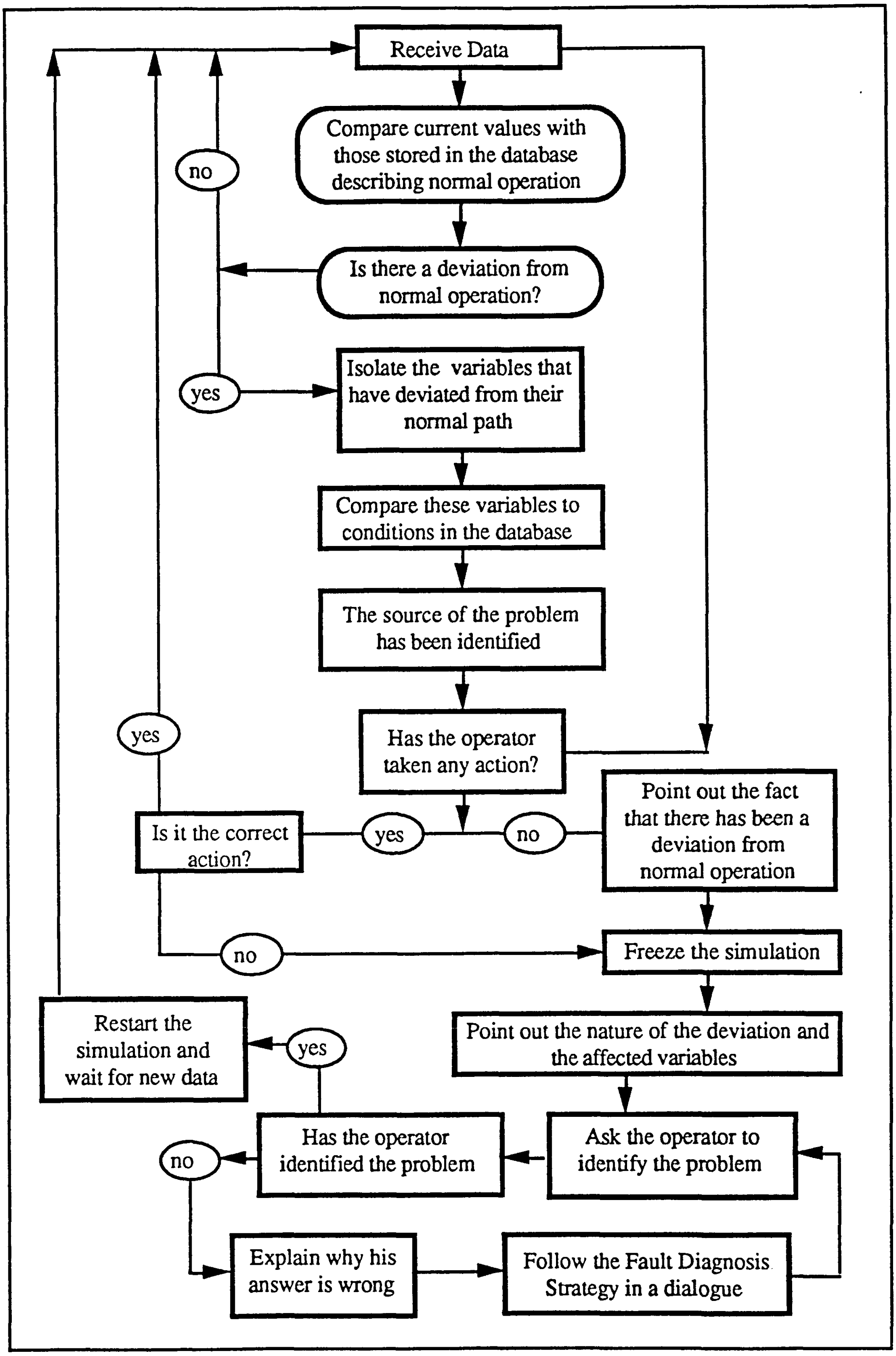


Figure 7.8. The Interactive Mode Fault Identification for the Evaporator Pilot Plant

Fault Identification Algorithm

The broad characteristics of the fault diagnosis algorithm that has been applied to the evaporator pilot plant application were outlined in section 4.1.4. The algorithm is able to recognise which fault out of a predefined set of possible faults has occurred while the plant was operating around a steady state. As previously mentioned this algorithm will receive data from RTPMS (current values of the variables displayed to the trainee, the status of RTPMS switches, etc) which it will then process using the knowledge base *heuristics*.

Two different types of information are used by the *heuristics*. The first type relates to the status of individual low level units in the plant structure. Thus the algorithm can directly identify if a manual valve has been CLOSED, if a pump has been switched OFF, etc. The second type of information used by the heuristics relates to the dynamic profiles of the displayed variables.

In the pilot plant application careful study of the results obtained by running the simulation problems of the complete plant (around a steady state) under normal and abnormal conditions showed that the introduction of a disturbance to the system affects only a distinct subset of the variables. Furthermore, it was realized that it is possible to broadly monitor the plant behaviour using the actual current value and two associated parameters which describe the status of each variable under normal operating conditions. These two parameters are: (a) the rate of change of each variable, and (b) the percentage change of each variable.

Obtaining the current value of a variable is straight forward since it is directly available from the SBTS. The evaluation of the rate of change of the variable is, however, slightly more complicated. A test algorithm has been implemented in RTPMS which evaluates the rate of change of each variable. A sample three term finite difference equation which estimates the rate of change of a Temperature (T) with respect to time (t) is presented below:

$$\left[\frac{\Delta T}{\Delta t}\right]_{est} = \frac{3T^{(k)} - 4T^{(k-1)} + T^{(k-2)}}{2\Delta t}$$

The algorithm uses three successive values of the variable (the current one $T^{(k)}$, and the two previous ones, $T^{(k-1)}$ and $T^{(k-2)}$) and estimates the rate of change over an augmented time interval. The same type equation can be used to estimate the rate of change of all the variables which are displayed to the trainee through the RTPMS interface (temperatures, flowrates, concentrations, and level). Finally, a simple equation was implemented in RTPMS to estimate the percentage change (increase or decrease) of these variables with respect to their previous value over a certain period of time.

These parameters are fed to the fault diagnosis sub-module where they are treated by a screening algorithm (figure 7.9.) which aims to identify any deviation from normal operation. The comparison is essentially performed using the two rate parameters (table 7.8). During normal operation (partial steady state with no plant upsets) the rate of change of each variable is approximately constant (or follows a well defined profile). Furthermore, the percentage change exhibited by each variable is dependent upon the rate of change and so is again approximately constant.

This is not the case under abnormal situations. An introduction of a disturbance in the system (for example switching off a pump, closing a manual valve, introducing a leak, etc.) affects the value and/or the dynamic profile of some of the system variables. For some variables the resulting effect is a percentage change leading to some new 'constant value', while for other variables there is a change in the dynamic profile (for example while it was gradually increasing it suddenly begins to decrease or there is a distinct change in the affected rate i.e. the rate of change either increases or decreases).

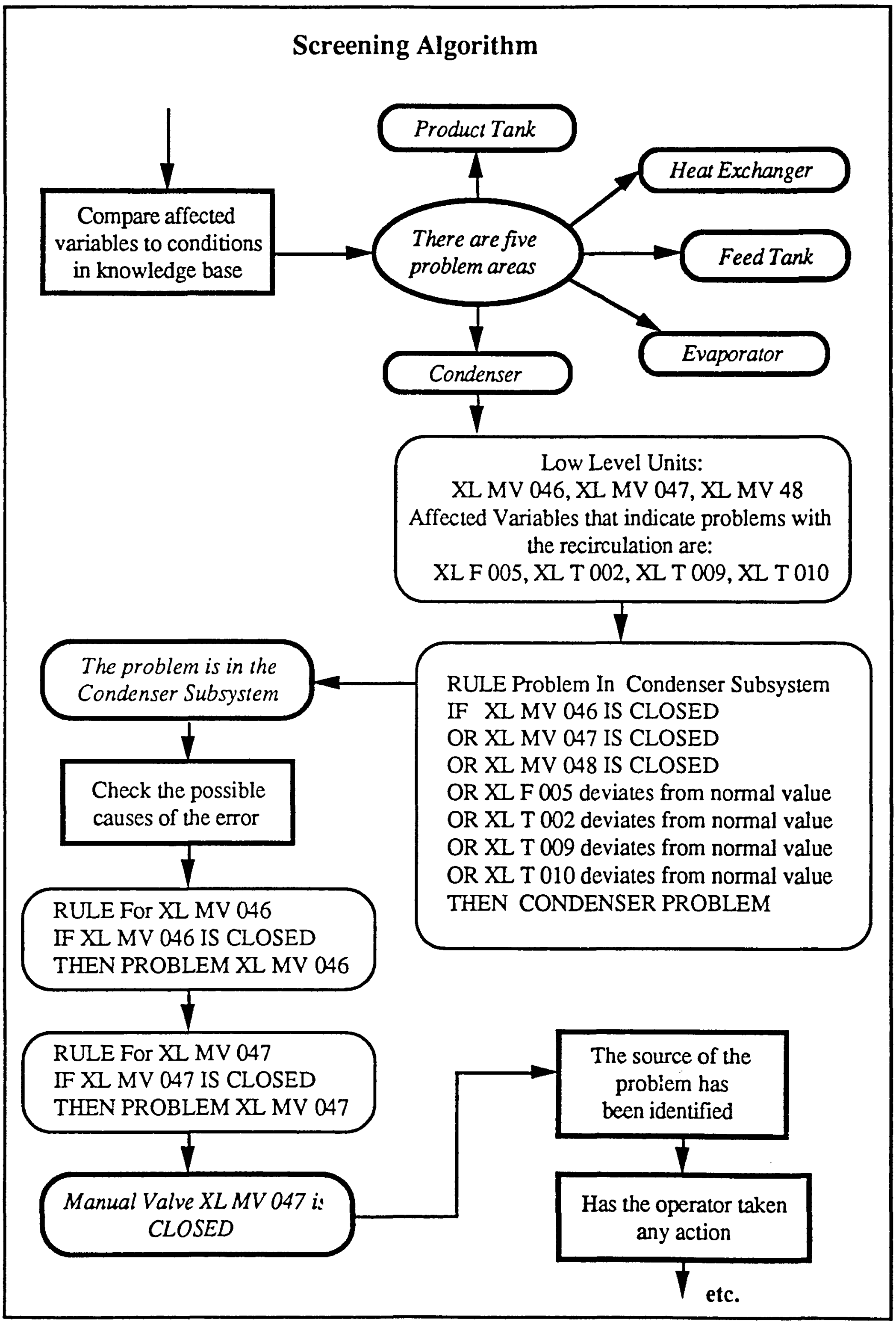


Figure 7.9. The Fault Identification Algorithm for the Evaporator Pilot Plant

Table 7.8. A rule used to identify a variable which deviates form its normal operation value

RULE for detecting deviation of XL F 005

IF XL F 005 rate of change > upper normal rate of change value

OR XL F 005 rate of change < lower normal rate of change value

OR XL F 005 percentage change > upper normal percentage change value

OR XL F 005 percentage change < lower normal percentage change value

THEN XL F 005 deviates from normal value

Having identified the fault the algorithm then waits for a prespecified time period and then checks whether the operator has taken any corrective action. This is achieved by the examination of a new set of data from RTPMS. For example if the disturbance was caused by a valve being shut the algorithm will be able to establish if the operator has opened it, etc. In this case it will inform the student that he has successfully dealt with the particular fault scenario and it will then direct him back to the *main fault identification* knowledge base. If however, no action or an incorrect action has been taken then the rules of the knowledge base will pause the execution of the simulation and the appropriate instructional knowledge base describing the problem subsystem will be activated.

7.5. Summary

The specific application described in this chapter demonstrates the general theme of this thesis. It establishes the feasibility of implementing both the integrated CBT system and the training programmes. It has been shown that it is possible to use different general purpose modules to easily design and configure flexible CBT systems with advanced functionalities and to implement comprehensive CBT programmes which are easy to support and update. Furthermore, the suitability of *Process Trainer* as the intelligent module of the integrated system architecture has been established.

It was also shown that the use of *Process Trainer* within the proposed system architecture permits the effective integration of training simulator and CAI/ICAI techniques. Moreover, the generic structure of *Process Trainer* which permits its easy adaptation to different applications has been demonstrated. It was thus shown that the tools and the formal methodologies outlined, can be applied to adapt, upgrade, or modify the current implemented application, or build a new functional system and define different training programmes.

It should be stressed that the development performed for the illustrative application was in depth rather than in breadth. Nevertheless, the complete structures of the individual system components have been implemented. As a result the completion of the training programmes for the illustrative application would only involve the inclusion of the missing domain specific information and the development of specific training sessions.

Finally, it is pointed out that the fault identification algorithm has only been used to exhibit the capabilities of *Process Trainer*. Therefore, while its structure has also been implemented its effectiveness has not been extensively tested. Its completion as a fullyfledged fault diagnosis system would involve considerably more effort than the rest of the components of the overall configuration (the SBTS and *Process Trainer)*. Nevertheless since it has been developed in a separate knowledge base it is possible to completely replace it with newly developed fault diagnosis techniques and methodologies. In this case the general rules described in chapter 6 could allow for their easy implementation.

Chapter 8

CONCLUSIONS

In this thesis the design of a system architecture is presented. This architecture provides tools which allow the easy development and implementation of flexible and easy to maintain CBT systems and courseware for both basic principles and 'hands-on' plant operation. Furthermore, several methodologies have been presented which utilize the tools offered by the system architecture to produce comprehensive CBT programmes for a range of training purposes in the process industries. Finally, a Training Demonstrator was developed that utilizes the tools and methodologies in a variety of ways for a specific application which involves an evaporator pilot plant.

The thesis began with a review of the requirements for operator training in the process industries and an examination of the role of the human operator in process operations. This indicated that training systems and courseware are needed which address the different types of knowledge, wide range of skills and range of cognitive activities which operators employ.

Additionally, the evolving role of the process operator leads to increasingly complex and sophisticated training requirements. As a result the need was identified for diverse and sophisticated training methods and training delivery platforms that can accommodate a variety of needs, ranging from background theory and 'hands-on' training in plant operation, to decision making and problem solving methodologies. These should also be flexible and easily modifiable in order to accommodate the changing role of the process operator.

A review of the state of the art in operator training technology then identified a sequence of pre-training steps that help formulate the training programmes. The advantages that computer based training has over traditional training methods were also reviewed. These showed that CBT provides the appropriate training medium with respect to meeting the requirements outlined above.

A detailed survey of the current state of the art in computer based training revealed that a great range of CBT courseware have been developed. However, their routine use especially with respect to training simulators is limited due to the costs involved in the initial development and subsequent maintenance and modifications of CBT courseware. Additionally, a need was identified for stand-alone CBT systems that could be used on the factory floor and of systems which are capable of providing on-line criticism and support to enhance 'hands-on' training.

Given the above needs, a novel integrated architecture for computer based training of process operators was formulated. An integrated system which is applicable to a wide range of training purposes in the process industries was then designed and implemented. This achieves advanced operator training functionalities while minimizing the user development effort through the integration of high level, general purpose modules.

The design uses, as far as possible, existing specially suited general purpose components as building blocks: a commercial dynamic simulator, an advanced plant control system, a flexible supervisory system for planning and management of discrete operations and an expert system module for the definition of operating procedures and training scenarios, the provision of advice, and the coordination of the other components and of the training programmes. Such an approach permits the rapid development of new operator training applications as well as the easy maintenance and subsequent modification of the resulting CBT system. It should therefore greatly reduce the associated costs.

Within the system, a variety of plant models, operating procedures and training scenarios are easily defined. A range of training methods may be incorporated, including structured tutorials, guided steady-state and dynamic demonstrations of typical operations, fully interactive plant operation from a variety of control viewpoints, and training for fault diagnosis with provision of advice. Thus, the flexible use of qualitative and quantitative aspects and training support tools is achieved.

In the first instance this has been proven with the design and subsequent implementation of a Simulation Based Training System (described in sections 4.2.2. and 7.3.). It has been shown that by using generic engineering orientated building blocks (for modelling, simulation, control and operations management) one can produce a *training simulator* that is not only 'up to standards' with the current state-of-the-art in process operator technology but also offers added features and functionalities.

The flexibility of the proposed system architecture allows the trainer to either develop new models, design training scenarios, etc. in environments specially suited for such development or use facilities and models already available within the company. This minimizes the development and implementation effort (development time and software/hardware requirements) towards a computer based training system which offers a realistic operator interface and a realistic representation of the process dynamics.

The effectiveness of this approach has been demonstrated in section 7.3. where different portions of the overall evaporator pilot plant simulation model are extracted and incorporated into individual simulation problems which emphasize the dynamics and operational characteristics of plant subsystems.

These findings can have direct implication on spreading the use of *training simulators*. The increasing use of dynamic simulators in the process industries accompanied by the development of simulation models (for non-training purposes) make possible the use of a company's own technological and human resources to easily develop its own in-house training applications. Thus the need to contract specialized experts for such development is reduced. Finally, while the proposed architecture has been developed and implemented in view of training requirements it could also provide a platform for the examination of all aspects which are dependent on the process dynamics.

Up to date most CBT systems use either a qualitative or a quantitative approach while the optimum configuration of a computer based training system would combine both. In the proposed integrated system architecture this is achieved with the integration of an *intelligent module* and the Simulation Based Training System. This provides for the development and implementation of comprehensive stand-alone computer based training programmes.

The specifications and desired functionalities of the *intelligent module* were presented in chapter 4 which described the integrated architecture for computer based training systems. An off-the-shelf building block that would meet these requirements did not exist. Therefore, in order to establish the feasibility of the complete system architecture it was necessary to implement such a module. This led to the design, development and implementation of *Process Trainer: A Tutoring System for Operations Staff* (a prototype 'expert system shell' for process training applications) which attempts to meet these requirements.

It has been shown that *Process Trainer* supports the combination of the qualitative and quantitative approaches to training. It therefore provides the desired platform for implementing training sessions which provide verbal descriptions, support and advice, while at the same time it enables the use, as necessary, of real-time simulations of varying degrees of complexity.

It was possible to address the various training objectives by manipulating the general purpose modules of the system. This was made possible through the definition, within *Process Trainer,* of general rules that one could use to easily implement different training programmes.

Three training programmes each addressing a different operator training area form the instructional basis for *Process Trainer.* They employ a variety of training methodologies, are stand-alone while also permitting on-line trainer intervention, and deliver instruction through a combination of theory and practice. Furthermore, the formal methodological definition of *Process Trainer* permits the easy definition and integration of additional training programmes.

Two of the training programmes are a novel implementation of formalized training programmes implementing the structured approach suggested by Bainbridge (1990). The *Tutorial* aims to prepare an operator for plant operation. It first introduces the trainee to the various plant components and their characteristics. Then it builds on this knowledge to introduce the concept of plant subsystems and their respective dynamics and cause and effect relationships. Finally, it examines subsystem interactions until ultimately it examines the whole plant.

The *Plant Operation Training Session* is the second training programme. The emphasis here is on real-time operation. A demonstration mode permits focusing attention on important cause and effect relationships, as well as the presentation of formal operating procedures such as start-up and shut-down. Alternatively an interactive mode allows for actual 'hands-on' training in a real environment. Under this mode the trainee is presented with a set of objectives which have to be met.

A novel methodology for training in fault diagnosis which integrates the two approaches traditionally followed forms the basis for the third training programme which has been implemented within *Process Trainer*. The methodology was then formalized as the *Training for Fault Identification* training programme and has been implemented as a stand-alone CBT system.

The knowledge base structures implemented in *Process Trainer* allow for the generic implementation of these training programmes. Thereafter, their adaptation to a particular problem domain would involve the incorporation within its structure of the specific domain information. This has been demonstrated in chapter 7.

While the training programmes address different training areas, each with a variety of training objectives and each having a different set of characteristics their implementation in *Process Trainer* offers some advantages and characteristics which are common to all. In summary these are:

1. Stand-alone capability
2. Individualisation of the training process
3. Uniformity in the knowledge transfer
4. Choice between engineer and operator training
5. Effective use of qualitative and quantitative knowledge representations
6. Utilization of static, dynamic graphics in interactive and non-interactive displays
7. Hands-on training in plant operation
8. Realistic representation of plant dynamics
9. Easy initial development, maintenance and modification thus flexible and cost-effective
10. Easy retrieval of the students performance for evaluation purposes

A specific application was illustrated which involves an evaporator pilot plant. This provided the means of establishing the feasibility of implementing both the integrated computer based training system and the training programmes. However, while the complete functional structure of *Process Trainer* has been implemented, the development with respect to the application has been in depth rather than in breadth. Once the feasibility of producing an instructional session was established (such as a demonstration, an interactive experience, etc.) the development of further examples which would illustrate the same principles was not undertaken. Nevertheless once the missing domain information are included the application will be complete.

It was thus shown that it is feasible to use different general purpose modules to design and configure with significantly less development effort, flexible CBT systems with advanced functionalities and to implement comprehensive CBT programmes which are easy to support and update. Furthermore, the generic structures of the training programmes which are implemented within *Process Trainer* can easily be adapted to different problem areas. Otherwise, the tools available and the formal methodologies for building up a functional system which will permit the definition of different training programmes are outlined.

In summary, the main concerns of this research regarded the ease of model development and maintenance, the ease of definition of the training sequences, the scope and mix of training methodologies that could be implemented, and the flexibility in the development of new training programmes.

So far the experience with integration of specialised but quite different components into one integrated whole shows that the concept is not only viable but that quite sophisticated training solutions can be achieved this way. The broad requirements and CBT system functionalities presented in chapters 2 and 3 respectively are largely met by the integrated system and *Process Trainer*, as demonstrated by the illustrative application to the evaporator pilot plant. Further application development is now quick, since each of the underlying tasks can be carried out in a software environment most suited to it. The need for the pre-training steps described in chapter 3 for the definition of a training programme are still there, but the tools to implement it are available.

In conclusion, the process industries' requirements and demands as they relate to the training of process operators have been met through a novel design which allows the implementation of CBT systems and courseware for a range of training purposes in the process industries. Furthermore, it was shown that it is possible to formalize the definition of training programmes for process operators in view of their implementation as CBT.

Chapter 9

FURTHER WORK

The present thesis led to the identification of a number of areas where further research and development would be justified. These can be grouped into two main categories. One is the further development of the current application and the second involves further research and development in training technology tools that could be implemented in conjunction with the current system.

9.1. Further Development of *Process Trainer*

The main areas for further development of *Process Trainer* may be summarized in the following categories:

1. Full scale case study evaluation of *Process Trainer*
2. Improvement of the user interface
3. Improvement of communications between building blocks
4. Improvement of the reporting capabilities
5. Development and implementation of appropriate student models
6. Extension of the Fault Identification Training Session to provide general Fault Management Training (detection, diagnosis, rectification, compensation, etc.)

9.1.1. Full Scale Case Study Evaluation of *Process Trainer*

A full scale case study evaluation of the integrated system architecture and *Process Trainer* was outside the scope of the current research. While they have both been fully implemented, the illustrative application has not been fully developed. Therefore, the first step would involve the completion of the instructional material of the training programmes for the evaporator pilot plant.

Then the evaluation of the system could be performed in conjunction with the actual departmental pilot plant. It should be possible to compare the effectiveness of *Process Trainer's* instructional approach with that of more traditional training methods. The evaluation could be based on the assessment of the training abilities and knowledge gained by students following one of the two training methods.

9.1.2. Improvements to the User Interface

While the user interface implemented for the prototype application was sufficient for testing and development purposes there is room for significant improvements that would enhance the system's overall effectiveness as a training medium. In particular, the development of a sophisticated user interface was hindered by the capabilities of the individual 'building blocks'. The current user interface is defined (and limited) by the functionalities of RTPMS and LEVEL5.

Recent upgrades to these now offer new graphics capabilities that could greatly enhance the current user interface. For example, the capability of using a WINDOWS environment, hypertext functions, animation, etc. offered by a new version of LEVEL5 which has recently become available will greatly enhance *Process Trainer's* user interfaces. Already a partial upgrade of *Process Trainer* has been performed and the advantages are already apparent.

9.1.3. Improvement of Communications

Once again the various building blocks that were used in the implementation of the proposed system architecture defined the nature of the communication links possible between them. While there is no problem with the communication between the control system, the simulation and the operations management modules, there is presently a bottleneck between them (the simulation based training system) and *Process Trainer*.

Currently this link is achieved through a file transfer communication protocol between the IBM PS/2, which runs *Process Trainer,* and the IBM 4341 mainframe which runs the simulation based training system. This leads to inefficient communication between the two platforms. However, the capability of a direct link between the two platforms, supported by the relevant application programs would eliminate all sluggishness. Such a direct link is possible with a version of RTPMS which is currently being installed in the department. Similar advantages would result by a proper networking of all the modules. This is however determined by the capabilities of the modules involved.

9.1.4. Improvement of the Reporting Capabilities

Currently the logging and reporting within the implemented system architecture is done at the level of each building block. While there is a possibility of having a centralized training audit facility implemented within *Process Trainer,* this is not presently implemented. The sluggishness of communications between it and the simulation based training system would interfere with the effectiveness of the entire configuration as a training medium. However, once the communication problems are resolved then an efficient centralized logging and reporting facility can easily be implemented.

9.1.5. Development of Student Models

The development of an intelligent student module was outside the scope of this thesis. Nevertheless the functional structures of simple student models that correspond to the evaluation requirements set out in the description of each of the three training programmes have been implemented. If those are to be used then one ought to define the actual evaluation steps and incorporate themin the assessment sections of individual knowledge base structures.

However, it is also possible to develop and implement an intelligent student model which would dynamically adapt the instructional process to meet the requirements and characteristics of individual users. The current implementation of *Process Trainer* and the division of the instructional material into small independent units should make the selection of the relevant subjects and presentation methods by an intelligent student model relatively easy. However, one should decide whether the complete instructional process should be 'student model controlled' or be interactively controlled by the trainee. Nevertheless, in the latter case a student model could take over the instructional process during the evaluation session and thus make it an extension of the instructional process.

9.1.6. Extension of the Fault Identification Training System

The current *Fault Identification Training Session* is of limited capabilities. It serves two purposes. First it is used to prove the feasibility of developing such a system within the proposed integrated system architecture. Second it is used to present the fault diagnosis strategy which was described in section 5.4. However, the extent of its development and its sophistication are presently limited to that of training for the identification of individual faults that may occur following normal conditions (around a steady state).

Further work is required for the development, testing and implementation of the current fault diagnosis algorithm. Also the current configuration of *Process Trainer* allows for the fault identification algorithm knowledge base to be completely substituted. A new knowledge base could be created that contains other well tested fault diagnosis strategies.

Another area for improvement of the current set up is the inclusion of a module that will train operators in other aspects of fault management. A module could be developed that will train operators in fault detection, another to train them in fault handling, etc. All these could be independently developed and then integrated in the general structure of *Process Trainer.*

9.2. An Integrated Training Programme Authoring System

As shown in the previous chapters the suggested integrated architecture allows for the development of a range of CBT systems which are suited to a whole range of training needs and objectives. Furthermore, the modular approach achieves advanced functionalities while minimizing the development effort. However, the current implementation employing SpeedUp, RTPMS, SUPERBATCH and LEVEL5 *(Process Trainer)* requires the developer of the training system to be familiar with each of these packages.

Nevertheless, the system architecture and the general methodologies employed both in implementing the system architecture and the respective training programmes is general enough to be pursued in other environments with other software packages. It is suggested that an authoring environment be developed that enjoys the advantages and functionalities of the current set-up while making it possible to access pre-defined modules and sub-modules as those described in chapter 4. These should allow the easy definition of computer based training system for different applications from within one integrated environment.

For this purpose it is thought that a User Interface Management System (implemented on a workstation) which allows the input of specific domain knowledge in the intelligent module, allows the effective definition of both dynamic and steady state simulations, permits the definition of operational sequences, and also allows the configuration of simple control strategies in a control emulator module could form the basis for an effective authoring system.

GLOSSARY

Adaptive Program. A program which makes itself more suited to its user.

Algorithm. A precise description of how to solve some specific problem.

Artificial Intelligence (AI). The science of getting computers to do things which, if done by humans, would be said to be intelligent.

Authoring System. A domain-independent component of an ITS (or CAI) that allows the developer to enter specific domain knowledge into the tutor's knowledge base.

Backward Chaining. A control procedure that attempts to achieve goals recursively, first by enumerating antecedents that would be sufficient for goal attainment and second by attempting to achieve or establish the antecedents themselves as goals.

Basic Principles Training Simulators. These present the integrated plant dynamics without all the details of a real plant.

Black-Box Expert System. A procedure that generates correct behaviour over a range of tasks in the domain, but whose mechanism is inaccessible to the ITS.

Branching Program. A teaching program organised as a set of frames.

Cognitive Fidelity. The measure of correlation between the cognitive model and actual human problem strategy.

Cognitive Model. A representation of human cognitive processes in a particular domain.

Cognitive Skill. Consists of knowing what to do in a situation, and being able to remember the necessary background information.

Cognitive. Cognitive Psychology is the study of higher mental processes. It is common to distinguish between perceptual and peripheral processes (processes dealing with information from sensors such as the eye or ear) and more central cognitive processes which deal with more abstract representations. Thus reasoning is thought to be a cognitive task while colour discrimination is perceptual.

Communication Protocol. The format of communication between different computer programs.

Communications Module. It manages the overall human-computer interface to the ICAI system. It is the part of the system that specifies or supports the activities that the student does and the methods available to the student to do those activities.

Computer Assisted Instruction (CAI). Teaching using the computer as a means of tutorial instruction. The use of the computer as an interactive training medium through tutorials , simulations and drills and practice.

Computer Assisted Learning (CAL). Synonym for *CAI.*

Computer Assisted Training (CAT). Synonym for *CAI.*

Computer Based Training (CBT). A broad term covering *CAI* and *CMI.*

Computer Based Training System. The actual software and hardware that are used to deliver the training and/or manage the instructional process (CMI).

Computer Managed Training (CMT). Synonym for *CMI.*

Computer Managed Instruction (CMI). The use of a computer to direct students through their training and produce statistical reports on student performance or system utilization.

Computer Managed Learning (CML). Synonym for *CMI.*

Control Room. The place where the operator performs his process control subtasks.

Control System Module. The part of the *simulation based training system* and the *integrated system architecture* that acts as the process control system for the simulated system. It allows the configuration of process control strategies and provides the interface for user interaction with the simulated process.

Courseware. The set of programs and associated materials for a computer assisted learning course.

Declarative Knowledge. Knowledge represented as basic principles and facts of a domain. It is usually contrasted with knowing how to use the facts, that is, *procedural knowledge.*

Dialogue. The sequence of messages between the user and the computer.

Domain Model. All knowledge that describe a particular problem domain (facts, heuristics, simulations, procedures, etc.).

Domain Representation. The data structures which are used to describe the domain characteristics in a domain model.

Drill. A series of questions of the same sort for a student to practise on.

Dynamic Simulator Module. The part of the *simulation based training system* and the *integrated system architecture* in which the process or part of it is modeled, and the simulation is executed.

Emulation. A computer based system which is indistinguishable from the actual system being emulated. Emulation can be distinguished from simulation on the basis of appearance.

Engineer. A person which defines the control and operational sequences that are executed by the *operator.*

Evaluation. The attempt to determine the extent to which educational objectives have been attained.

Expert Model. Synonym for *domain model.*

Expert Module. The module of an ITS that provides the domain expertise, i.e., the knowledge that the ITS is trying to teach. The definition of the domain expertise is included in the *domain model.*

Expert System. A computer program that uses a knowledge base and inference procedures to act as an expert in a specific domain. It is able to reach conclusions very similar to those reached by a human expert.

Explanation. Motivating, justifying, or rationalizing an action by presenting antecedent considerations such as goals, laws, or heuristic rules that affected or determined the desirability of the action.

Expository Tutor. A tutor that is concerned with *declarative knowledge.* Usually interactive dialogue is the instructional tool used by this type of tutor.

Fault Detection. The realization that a fault has occurred in the process which has affected normal operation.

Fault Diagnosis. The recognition of the actual fault that has caused abnormal operational behaviour.

Fault Identification. Synonym for *fault diagnosis.*

Feedback. In general, a technique whereby part of the output of a system is redirected to the input; in education, the information provided to a learner about what he has done.

Fidelity. A measure of how closely the simulated environment matches the real world. There are three kinds of fidelity: physical, operational, and psychological.

Forward Chaining. A control procedure that produces new decisions recursively by affirming the consequent propositions associated within an inferential rule with antecedent conditions that are currently believed. As new affirmed propositions change the current set of beliefs, additional rules are applied recursively.

Frame. In teaching programs, one step consisting of some text to be presented to the student and a specification of how the next frame is to be determined by the student's response to the text; in AI, a data structure intended to describe some stereotypes situation.

Full Scope Training Simulators. Synonym for replica simulators.

Generative Program. A teaching program which produces questions, text, etc. from partial specifications while it is running.

Glass-Box Expert System. An expert system that contains human-like representation of knowledge. This type of expert system is more amenable to tutoring than a black-box expert system because it can explain its reasoning.

Heuristic Knowledge. Task-dependent knowledge that helps in achieving it. Includes actions taken by an expert to make measurements or perform transformations in the domain.

Heuristic. A piece of information that might be useful in solving a problem (often contrasted with *algorithm*).

High Fidelity Training Simulators. Either basic principle or full scope simulators that are a very close representation of the simulated system. It could be the complete process or part of the process.

High-Level Language. A programming language which permits a programmer to write programs without knowing the details of a particular computer's built-in instructions.

Highest Level Units. Large interdependent combinations of low and medium level units into plant subsystems.

Individualised Instruction. Instruction which is geared to the individual student's competence and aptitude (rather than those of a hypothetical average student in a group).

Instruction. Actual presentation of curriculum material to the student.

Instructional Process. The process and sequence of delivering the instructional material through a computer based training programme.

Instructional Strategies. The strategy employed in the training methods to deliver.

Integrated System Architecture (ISA). The complete system architecture which integrates all four modules (dynamic simulator, control system module, operations management system, and intelligent module).

Intelligent Computer Assisted Instruction (ICAI). A computer program that (a) is capable of competent problem solving in a domain, (b) can infer a learner's approximation of competence, and (c) is able to reduce the difference between its competence and the student's through application of various tutoring strategies.

Intelligent Module. The part of the *simulation based training system* and the *integrated system architecture* which delivers the qualitative instructional material, handles the execution of the training programmes, and co-ordinates the rest of the modules.

Intelligent Tutoring System (ITS). Synonym for Intelligent Computer Assisted Instruction.

Interface. A shared boundary, between two pieces of equipment, between two programs, or between a user and a computer.

Internal models. An internal representation of domain features and characteristics employed by the operators. *(mental models)*

Knowledge Base. The repository of knowledge in a computer system.

Learner Control. The means by which the learner determines (or helps determine) learning activities.

LEVEL5. A commercial expert system development environment.

Linear Program. A teaching program organised as a set of frames, each of which always goes on to the next frame.

Low Level Units. Plant components which have an implicit effect on the process, such as valves, controllers, transducers, etc.

Master Procedure. SUPERBATCH feature. A master procedure is a directed network of *resource phases* which define the overall processing task sequence to achieve an operational objective. Master procedures also include the values of parameters, processing times and other processing information to be used throughout the computer aided production management system which uses the computer based control system to automate the planning and control activities of batch processes.

Medium Level Units. Major unit operations that play a direct and important role in the process.

Mental Model. The operator's general background knowledge of how the process works, which is used in explaining what is happening, predicting what is going to happen, identifying unknown faults, etc. A different meaning also used is the operator's mental picture of the current, and immediate future, states of the process.

Misconception. An item of knowledge that the student has and the expert does not have.

Missing Conception. An item of knowledge that the expert has and the student does not have.

Natural Language. The conventional method for exchanging information between people, such as English as a means of communication for human speakers and various formal written systems as a means of representing intentions in technical disciplines.

Objectives. The aims or goals of instruction.

Operations Management Module. The part of the *simulation based training system* and the *integrated system architecture* in which operational sequences and procedures can be defined for automatic execution.

Operator. A person controlling and supervising the process from the *control room.*

Phase. SUPERBATCH feature. A phase is a group of processing tasks that take a batch at a given state and process it until 1) it reaches a safe or logical point where processing may be held in order to allow external supervisory intervention; 2) the subset of the plant resources required to produce the batch change.

Plant Subsystems. Large sub-parts of a plant which demonstrate characteristic dynamic behaviour. Made by a combination of low and medium level units. (also *highest level units*)

Plant Units. Independent sub-parts of the plant between which the flows are relatively simple. Three categories of units are used: low level units, medium level units, and highest level units.

Pre-Training Steps. A number of steps which help define the training programme. These are: identification of the trainee's characteristics, analysis of the training needs, identification of the training objectives, task analysis.

Procedural Knowledge. Domain-dependent knowledge about how to perform a specific task.

Procedural Tutor. A type of tutor that teaches procedural knowledge, i.e., skills and procedures. Usually exercises and examples are used by procedure tutors.

PROCESS TRAINER. A prototype expert system shell for process training applications. It acts as the *intelligent module* in the *integrated system architecture.*

Production Rule. A rule in which a condition is associated with an action to be carried out if the condition holds, of the form *condition(s) imply action(s).* A set of production rules and an interpreter for processing them is termed a *production system.*

Programmed Instruction. A method of teaching based on linear, and branching programs.

Qualitative Approach to Training. The use of text, diagrams, etc. to create dialogues that aim to present instructional material to students.

Qualitative Process Model. A type of cognitive model, concerned with reasoning about the causal structure of the world; the simulation of dynamic processes in the mind. It is an important facet of trouble-shooting behaviour.

Quantitative Approach to Training. The use of mathematical simulations to describe the instructional material. Supports the interactive 'hands-on' operation of a process and also indicates explicitly the cause and effect relationships that define a process system.

Reinforcement. In behaviourist theories of learning, the presentation of a reward after the desired response has been made.

Replica Training Simulators. A simulation of a system that mimics all aspects of a working system (process, control room panels, work environment, etc.)

Resource Phase. SUPERBATCH feature. The resource phase describes a processing task at the resource level, and therefore determines the subset of the *plant resources* required to carry out a given processing task. Resource phases are used in *master procedures* therefore eliminating the need for separate master procedures for specific unit allocations.

RTPMS. A commercial full scale distributed control system and process management system.

Rule-Based Model. An expert module of an ITS that is implemented with a rule-based (production) system. (Also called a "production model.")

Rule. A pair, composed of an antecedent condition and a consequent proposition, which can support deductive processes such as backward chaining and forward chaining.

Scheduling. Determining the order of activities for execution.

Simulation Based Training System (SBTS). The implemented system that serves as an advanced training simulator. It uses general purpose modules: a dynamic simulator, a control system module, and an operations management system.

Simulation. A mimic of part of a real-life system and which can be used by the student to learn about this system.

SpeedUp. A commercial process simulator.

Stand-Alone System. A system that can be used for learning and/or deliver instruction independently without a human tutor.

Student Model. That part of a teaching program which provides it with information about the student using it. The component (a data structure) of an ITS that represents the student's current state of knowledge (mastery) of the domain, i.e., a detailed model of student cognition.

Student Module. Represents and assesses the user's understanding or progress in understanding the domain model. The knowledge structure that depicts the student's current state is the *student model.*

Student-Expert Differences. The difference between the expert's knowledge and the student's knowledge. There are two basic types of differences; missing conceptions and misconceptions. The three models to represent student-expert differences are: overlay model, bug library technique, and library of bug parts.

SUPERBATCH. A commercial operations management system.

Task Analysis. The process of dividing and subdividing jobs into tasks, and tasks into steps.

Training Delivery Platforms. The medium and means used to deliver the training material. In the case of CBT, actual CBT system.

Training Method. The method employed to deliver the instructional material to a student.

Training Programme. A formalized instructional process which aims to address a specific set of *training objectives.*

Training Simulator. A computer simulation of a given system.

Tutor Module. It controls how and in what order the domain concepts should be introduced, and monitors the student's progress in solving domain problems. It includes the tutor model.

User Interface Management System (UIMS). A strategy that attempts to separate the interface component of an application program from the computational part.

User Interface. The interface between the user and the computer program.

Verbal Protocol. A protocol which verbally describes the execution of particular tasks. Usually operators are requested to think aloud so that a record of their thoughts along with their actual manual actions can be made.

REFERENCES

Alty J.L., Elzer P., Holst O., Johansen G. and Savory S., (1986). 'Literature and User Survey of Issues Related to Man-Machine Interfaces for Supervision and Control.' Scottish Human Computer Interaction Centre, ESPRIT: P600, Report No. AMU8603/01S.

Ancarani A. and Zanobetti D. (1983). 'Simulators for Nuclear Power Plants.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 3:06.

Anderson J.R. (1988). 'The Expert Module.' In: *Foundation of Intelligent Tutoring Systems* (Eds. M.C. Polson and J.J. Richardson), Lawrence Erlbaum Associates Publishers, New Jersey, U.S.A. pp 21-53.

Andow P.K., (1984). 'Expert Systems in Process Plant Fault Diagnosis.' In: *Ergonomics Problems in Process Operations.* Institution of Chemical Engineers Symposium Series No. 90. Pergamon Press. pp 49-58.

Atkinson R.C. (1976). 'Adaptive Instructional Systems: Some Attempts to Optimize the Learning Process.' In: *Cognition and Instruction* (Ed. D. Klahr). Lawrence Erlbaum Associates, Inc. New Jersey, U.S.A. pp 81-108.

Bainbridge L. (1974). 'Analysis of Verbal Protocols from a Process Control Task.' In: *The Human Operator in Process Control*; (Eds. E. Edwards and F. P. Lees). Taylor & Francis Ltd. London. pp 146-158.

Bainbridge L. (1990). 'A Note on Training Sequences and Simulator Facilities for Industrial Process Operation.' Paper presented at PIRTEM Cognitive Approaches to Supervisory Control in Dynamic Environments' Network Meeting on Operator Training and Acquisition of Cognitive Skills. School of Psychology, University of Wales College of Cardiff, September 27-28.

Bainbridge L., Beishon J., Hemming J.H. and Splaine M., (1974). 'A Study of real-time decision-making using a plant simulator.' In: *The Human Operator in*

Process Control; (Eds. E. Edwards and F. P. Lees). Taylor & Francis Ltd. London. pp 91-110.

Baker S.and Marshall E. (1989). 'Simulators for Training and the Evaluation of Operator Performance.' In: *Developing Skills with Information Technology* (Eds. L. Bainbridge and S.A. Quintanilla. Wiley. pp 293-314.

Begg A.G. and Weldon R.A., (1983). 'Computer Based Training Aids.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol May 1982*. The Institution of Nuclear Engineers, London. Paper No. 2:06.

Bernal R.A., (1986). *The Development of a Dynamic Model for a Pilot Plant Evaporator*, M.Sc. Thesis Department of Chemical Engineering and Chemical Technology of Imperial College of Science and Technology.

Billing P.J, Yue P.L. and Treadgold T., (1986). 'Open learning on the factory floor.' *The Chemical Engineer*, July/August 1986. pp 24-25.

Billing P.J. and Yue P.L. (1985). 'Computer Simulation of Plant Operation.' In: *Process Systems Engineering: PSE 85*. The Institution of Chemical Engineers, London. pp 251-261.

Billing P.J. and Yue P.L. (1987). 'Modelling and Simulation of Unit Operations for Use in Process Operator Training.' In: *Proceedings of the XVIII Congress: The Use of Computers in Chemical Engineering*, Giardini, Italy.

Black G.C.J. (1987). 'Training: Key to Staying on Track.' *Chemical Engineering*, July 1987. pp 107-110.

Bradbury B.A. and Brooks R.W. (1983). 'The Advanced Control System.' In: *I.Chem.E. Symposium Series No. 79*, Pergamon Press, London. pp 220-225.

Brouwers A.A.F., (1984). 'Automation and the Human Operator; Effects in Process Operations.' In: *Ergonomics Problems in Process Operations* Institution of Chemical Engineers Symposium Series No. 90. Pergamon Press, London. pp 179-191.

Brown J.S, Burton R.R. and Bell A.G. (1975). 'SOPHIE. A Step towards a reactive learning environment.' *International Journal of Man Machine Studies,* 7. pp 675-696.

Brown J.S., Burton R.R. and deKleer J. (1982). 'Pedagological, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II and III.' In: *Intelligent Tutoring Systems* (Eds. D.H. Sleeman and J.S. Brown), Academic Press, London. pp 227-282.

Burns H.L. and Capps C.G. (1988). 'Foundations of Intelligent Tutoring Systems: An Introduction.' In: *Foundation of Intelligent Tutoring Systems* (Eds. M.C. Polson and J.J. Richardson), Lawrence Erlbaum Associates Publishers, New Jersey, U.S.A. pp 1-19.

Burton R.R. (1988). 'The Environment Module of Intelligent Tutoring Systems.' In: *Foundation of Intelligent Tutoring Systems* (Eds. M.C. Polson and J.J. Richardson), Lawrence Erlbaum Associates Publishers, New Jersey, U.S.A. pp 109-142.

Burton R.R. and Brown J.S. (1982). 'An Investigation of Computer Coaching for Informal Learning Activities.' In: *Intelligent Tutoring Systems* (Eds. D. Sleeman and J.S. Brown), Academic Press, London. pp 79-98.

Charniak E. and McDermott D. (1985). *Introduction to Artificial Intelligence.* Addison-Wesley Publishing Company.

Chemical Week (1983). 'Training Plant Operators.' September 21. pp 50-53.

Clancey W.J. (1982). 'Tutoring Rules for Guiding a Case Method Dialogue.' In: *Intelligent Tutoring Systems* (Eds. D.H. Sleeman and J.S. Brown), Academic Press, London. pp 201-225.

Clancey W.J. (1988). 'The Role of Qualitative Models in Instruction.' In: *Artificial Intelligence And Human Learning: intelligent computer-assisted instruction* (Ed. J. Self). Chapman and Hall Ltd, London. pp 49-68.

Cott B.J. (1989). *An Intelligent Computer-Aided Production Management System for Batch Chemical Processes.* PhD Thesis Imperial College of Science, Technology and Medicine. University of London.

Cott B.J. and Macchietto S. (1989). 'An Integrated Approach to Computer-Aided Operation of Batch Chemical Plants.' *Comput. Chem. Engng.*, **13**. pp 1263-1271.

Dartt S.R. (1985). 'A Survey on Process Control Applications Needs.' *Chemical Engineering Progress*, December 1985. pp 11-14.

Dean C. and Whitlock Q. (1988). *A Handbook of Computer Based Training* (2nd Ed.). Kogan Page Ltd. London, U.K.

Duncan K.D. (1974). "Analytical Techniques in Training Design.' In: *The Human Operator in Process Control*; (Eds. E. Edwards and F. P. Lees). Taylor & Francis Ltd. London.

Eberts R.E. (1985). 'Cognitive Skills And Process Control.' *Chemical Engineering Progress*, December. pp 30-34.

Eberts R.E. and Brock J.B. (1987). 'Computer-Assisted and Computer-Managed Instruction.' In: *Handbook of Human Factors* (Ed. G. Salvendy). New York: Wiley. pp 976-1011.

Elston Y. and Potter D. (1989). 'Simulator Trains for New Equipment Use.' *Hydrocarbon Processing* 68(12). pp 85-90.

Fielden J. (1979). 'The Cost of CAL.' In: *Learning Through Computers* (Ed. D.A. Tawney). The MacMillan Press Ltd, London. pp 159-170.

Flexman R.E. and Stark E.A. (1987). 'Training Simulators.' In: *Handbook of Human Factors* (Ed. G. Salvendy). New York: Wiley. pp 1012-1038.

Frost D. (1987). 'A Natural Language Interface for Expert Systems: System Architecture.' In: *Advances in Artificial Intelligence* (Proceedings of the 1987 AISB Conference, University of Edinburgh, 6-10 April 1987), (Eds. J. Hallam and C. Mellish). John Wiley & Sons. pp 157-168.

Geary D. (1988). 'Pioneering Open Learning for BTEC Qualifications.' Paper presented at the I.Ch.E. conference on *The Selection Training and Motivation of Process Operators* held in Kingston upon Hull on March 2, 1988.

Genesereth M.R. (1982). 'The Role of Plans in Intelligent Tutoring Systems.' In: *Intelligent Tutoring Systems* (Eds. D.H. Sleeman and J.S. Brown), Academic Press, London. pp 137-155.

Gevarter W.B. (1987a). 'Introduction to Artificial Intelligence.' *Chemical Engineering Progress*. Vol **83**, No. 9, September. pp 21-37.

Gevarter W.B., (1987b). 'The Nature and Evaluation of Commercial Expert System Building Tools.' *Computer.* May. pp 24-41.

Gilmore D. and Self J. (1988). 'The Application of Machine Learning to Intelligent Tutoring Systems.' In: *Artificial Intelligence and Human Learning: intelligent computer-assisted instruction* (Ed. J. Self). Chapman and Hall Ltd. New York, U.S.A.

Gnosspelius C.F. (1983). 'Training for Nuclear Power Plant Operation in Sweden.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 1:02.

Goldstein I.P. (1982). 'The Genetic Graph: a Representation for the Evolution of Procedural Knowledge.' In: *Intelligent Tutoring Systems* (Eds. D.H. Sleeman and J.S. Brown), Academic Press, London. pp 51-77.

Gran E., Malik T.I. and Naess L. (1988). 'Process Systems Engineering Opportunities Presented through Real Time, Full Scale Simulation of Plants.' In: *Process Systems Engineering '88* (Eds. M.L. Brisk and J.D. Perkins). The Institution of Engineers, Australia, NCP 88/17. pp 270-275.

Gregg L.W. (1976). 'Methods and Models for Task Analysis in Instructional Design.' In: *Cognition and Instruction* (Ed. D. Klahr). Lawrence Erlbaum Associates, Inc. New Jersey, U.S.A. pp 109-116.

Hack K.M. (1988). 'Why train process operators.' Paper presented at the I.Ch.E. conference on *The Selection Training and Motivation of Process Operators* held in Kingston upon Hull on March 2, 1988.

Halff H.M. (1988). 'Curriculum and Instruction in Automated Tutors.' In: *Foundation of Intelligent Tutoring Systems* (Eds. M.C. Polson and J.J. Richardson), Lawrence Erlbaum Associates Publishers, New Jersey, U.S.A. pp 79-108.

Hamilton J., Hacking D., McWhirter A.F., and Wiltshire J. (1983). 'The Design of the Hunterston 'B' AGR Training Simulator.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 2:03.

Harmon P. (1988). 'Currently Available Expert System-Building Tools.' *Expert Systems Strategies*. Vol. **4**. No. 7. pp 3-15.

Hayes-Roth F.W., Waterman D.A. and Lenat O.B.(1983) Editors. *Building Expert Systems*. Addison-Wesley London.

Helget A.G., Britt H.I., Boston J.F., and Evans L.B. (1990). 'The Architecture for a Next Generation Dynamic Process Simulator.' Prepared for presentation at November 1990 AIChE Meeting "New Concepts in Dynamic Simulation" Session.

Herman D.J., Sullivan G.R., and Thomas S. (1985). 'The Integration of Process Design, Simulation, and Control Systems.' In: *Process Systems Engineering: PSE 85*. The Institution of Chemical Engineers, London. pp 505-516.

Holding D.H. (1987). 'Concepts of Training.' In: *Handbook of Human Factors* (Ed. G. Salvendy). New York: Wiley. pp 939-962.

Hollan J.D. (1984). 'Intelligent Object-Based Graphical Interfaces.' In: *Human-Computer Interaction* (Ed. G. Salvendy). Elsevier Science Publishers B.V., Amsterdam. pp 293-296.

Hollan, Hutchins, and Weitzman (1984). 'Steamer: An Interactive Inspectable Simulation Based Training System. *AI Magazine,* **5**. pp 15-27.

Hutchison H.P., Jackson D.J., and Morton W. (1986). The Development of an Equation-Oriented Flowsheet Simulation and Optimization Package.' *Computers Chem. Engng*. Vol. **10**. pp 19-47.

Information Builders Inc. (1987). *Level 5 Expert System Software Users Manual.*

Johnson W.B. (1988). 'Pragmatic Considerations in Research, Development, and Implementation of Intelligent Tutoring Systems.' In: *Foundation of Intelligent Tutoring Systems* (Eds. M.C. Polson and J.J. Richardson), Lawrence Erlbaum Associates Publishers, New Jersey, U.S.A. pp 191-207.

Jones D.R. and Brook J. (1990). "High Fidelity Real Time Simulation and its Application to Olefins Plant Operator Training". In: *Computer Applications in Chemical Engineering* (Proceedings of the European Symposium on Computer Applications in Chemical Engineering, ComChem '90, The Hague, The

Netherlands, May 7-9, 1990), (Eds. by H.Th. Bussemaker and P.D. Iedema). Elsevier Science Publishers. Amsterdam, The Netherlands. pp 217-222.

Kassianides S. and Macchietto S. (1990). 'A Simulation Based System for Operator Training.' In: *Computer Applications in Chemical Engineering* (Proceedings of the European Symposium on Computer Applications in Chemical Engineering, ComChem '90, The Hague, The Netherlands, May 7-9, 1990), (Eds. by H.Th. Bussemaker and P.D. Iedema). Elsevier Science Publishers. Amsterdam, The Netherlands. pp 203-208.

Kershenbaum L. and Macchietto S. (1984). 'The IBM Project for Chemical Engineering Education at Imperial College.' *I.Chem.E. Symposium Series 101.* Pergamon Press, London. pp 307-321.

Kletz T.A. (1985). *An Engineer's View of Human Error.* Institution of Chemical Engineers.

Kletz T.A. (1985). *What Went Wrong? Case Histories of Process Plant Disasters.* Gulf Publishing Company.

Kletz T.A. (1988). 'Mistakes Operators Make.' Paper presented at the I.Ch.E. conference on *The Selection Training and Motivation of Process Operators* held in Kingston upon Hull on March 2, 1988.

Koppel L. B. and Sullivan G.R. (1986). 'Use of IBM's Advanced Control System in Undergraduate Process Control Education.' *Chemical Engineering Education,* Spring 1986. pp 70-73, 106-107

Kroner A., Holl P., Marquardt W., and Gilles E.D. (1990). 'DIVA-An Open Architecture for Dynamic Simulation.' *Computers chem Engng,* Vol. **14**, No. 11. pp 1289-1295.

Lakey J.R.A. and Gibbs D.C.C. (1983). 'Training for Operating Personnel for Nuclear Ships.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 1:03.

Leach A. R. D. (1988). Seminar given at the CAPE Meeting at the Rutherford Appleton Laboratory on the "Use of Workstations in Process Engineering" on January 29.

Leitch R. and Horne K.D. (1988). 'Prospects for Intelligent Training Systems in Industrial Environments.' ESPRIT: P2615.

Lhoir J.L. (1984). 'Selection and Training of Nuclear Power Plant Operators (EC Countries, Switzerland, Spain, Sweden).' Commission of the European Communities, Report EUR 7769 EN.

Li-Chi Cliff Po (1988). 'PCTRAN: a transient analysis code for personal computers.' *Nuclear Engineering International,* May 1988. pp 47-50.

Luder Th., Wonzy G. and Jeromin L. (1990). 'The Application of an Operator Training System in Education.' In: *Computer Applications in Chemical Engineering* (Proceedings of the European Symposium on Computer Applications in Chemical Engineering, ComChem '90, The Hague, The Netherlands, May 7-9, 1990), (Eds. by H.Th. Bussemaker and P.D. Iedema). Elsevier Science Publishers. Amsterdam, The Netherlands. pp 209-216.

Lysaught J.P. and Williams C.M. (1963). *A Guide to Programmed Instruction.* John Wiley & Sons Inc. U.S.A.

Macchietto S., Matzopoulos M., and Stuart G. (1987). 'On-Line Simulation and Optimization as an Aid to Plant Operation - The Easy Way.' In: *Proceedings of the Foundations of Computer-Aided Process Operations* (FOCAPO 1987) (Eds. Reklaitis G.V. and Spriggs H.D.), Park City, UTAH, USA, July 5-10. pp 669-679.

Marshall E.C., Scanlon K.E., Shepherd A. and Duncan K.D. (1981). 'Panel Diagnosis Training for Major-Hazard Continuous-Process Installations.' *The Chemical Engineer,* February. pp 66-69.

Marshall G. (1988). 'Interactive Video Training in the Workplace.' Paper presented at the I.Ch.E. conference on *The Selection Training and Motivation of Process Operators* held in Kingston upon Hull on March 2.

McCaughey F. and Geary D. (1988). 'The Open Learning Course for Process Operators.' Paper presented at the I.Ch.E. conference on *The Selection Training and Motivation of Process Operators* held in Kingston upon Hull on March 2.

McFarlane A.F. (1983). 'Computer Aided Instruction in the Nuclear Training Field.' In: *Proceedings of an International Conference Training For Nuclear Power Plant*

Operation, Bristol. May 1982. The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 1:05.

Miller J.R. (1988). 'The Role of Human-Computer Interaction in Intelligent Tutoring Systems.' In: *Foundation of Intelligent Tutoring Systems* (Eds. M.C. Polson and J.J. Richardson), Lawrence Erlbaum Associates Publishers, New Jersey, U.S.A. pp 143-189.

Moore R.L. and Kramer M.A. (1985). 'Expert Systems in On-Line Process Control.' *CPC* III. pp 839-867.

Myerscough P.B. and Hignett E.T. (1983). 'On and Off-Site Training of Technical Staff in the United Kingdom Nuclear Power Stations.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 4:04.

Nedderman J., (1988). 'Can Computers Replace Power Plant Operators?.' *Nuclear Engineering International,* May 1988. pp 40-41.

Niida K., Itoh J., Umeda T., Kobayashi S., and Ichikawa A. (1986). 'Some Expert System Experiments in Process Engineering.' *Chem. Eng. Res. Des.* Vol. **64**. September. pp 372-380.

Nuclear Engineering (1988). 'Westinghouse, Framatome and EdF join forces in CAT Co.' May. pp 46-47.

O'Shea T. and Self J. (1988). *Learning and Teaching with Computers: Artificial Intelligence in Education.* Harvester Press. Brighton, England.

O'Shima E. (1983). 'Computer Aided Plant Operation.' *Computers and Chemical Engineering* Vol. **7**, No. 4. pp 311-329.

Padden K., Macaulay G., and Thorson J. (1983). 'Computer Based Training for The Nuclear Industry.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 2:02.

Pantelides C.C. (1988). 'SpeedUp - Recent Advances in Process Simulation.' *Computers and Chemical Engineering,* **12**. pp 745-755.

Patrick J. and Haines B. (1988). 'Training and Transfer of Fault-Finding Skill.' *Ergonomics* Vol. **31**, no.2., pp 193-210.

Payne A. Hutchings B. and Ayre P (1980). *Computer Software for Schools.* Pitman Press, Bath, U.K.

Payne S.J. (1988). 'Methods and Mental Models in Theories of Cognitive Skill.' In: *Artificial Intelligence and Human Learning: intelligent computer-assisted instruction* (Ed. J. Self). Chapman and Hall Ltd. New York, U.S.A.

Perkins J.D. and Barton G.M. (1987). 'Modelling and Simulation in Process Operation.' In: *Proceedings of the Foundations of Computer-Aided Process Operations* (FOCAPO 1987) (Eds. Reklaitis G.V. and Spriggs H.D.), Park City, UTAH, USA, July 5-10. pp.287-316.

Perris F.A. (1990). 'The Growing Importance of Dynamic Simulation for Process Engineers.' In: *Dynamic Simulation in the Process Industries,* Symposium Papers 1990 No.4. UMIST, Manchester. June 20. Paper No. 1.1.

Peterson J.W.M. (1979). 'Package Design: Tutorial Dialogue.' In: *Learning Through Computers,* (Ed. D.A. Tawney). The MacMillan Press Ltd, London. pp 130-140.

Postlethwaite B. (1987). 'Continuing Education in Process Control using ACS.' *The Chemical Engineer,* April 1987. pp 17-18.

Process Engineering (1985). 'Integrating Process Control and Production Management.' July. pp 40-41.

Process Engineering (1987). 'Classroom on the Small Screen.' November. pp 47-48.

Quintanilla S.A. (1989). 'Intelligent Tutoring Systems (ITS) in Training.' In: *Developing Skills with Information Technology* (Eds. L. Bainbridge and S.A. Quintanilla. Wiley. pp 329-338.

Resnick L.B. (1976). 'Task Analysis in Instructional Design: Some Cases from Mathematics.' In: *Cognition and Instruction* (Ed. D. Klahr). Lawrence Erlbaum Associates Inc. New Jersey, U.S.A. pp 51-80.

Rozo P.R. (1987). *Development of a Real-Time Simulation for a Forced Recirculation Evaporator.* M.Sc. Thesis Department of Chemical Engineering and Chemical Technology of Imperial College of Science and Technology.

Sachs P.A., Paterson A.M. and Turner M.H.M. (1986). 'ESCORT - an Expert System for Complex Operations in Real Time.' *Expert Systems*, Vol.3, No. 1, January 1986. pp 22-29

Shepherd A. (1985). 'Hierarchical Task Analysis and Training Decisions.' *Programmed Learning and Educational Technology, 22.* pp 162-176.

Shepherd A. (1986). 'Issues in the Training of Process Operators.' *International Journal if Industrial Ergonomics, 1.* pp 49-64.

Shepherd A. (1989). 'Issues in Training.' Personal Communication.

Shepherd A., Marshall E.C., Turner A. and Duncan K.D. (1977). 'Diagnosis of Plant Failures from a Control Panel: A Comparison of Three Training Methods.' *Ergonomics.* Vol. 20, No. 4. pp 347-361.

Shinohara T. (1987). 'A Block Structured Approach to Dynamic Process Simulation.' In: *Proceedings of the Foundations of Computer-Aided Process Operations* (FOCAPO 1987) (Eds. Reklaitis G.V. and Spriggs H.D.), Park City, UTAH, USA, July 5-10. pp 317-331.

Sleeman D. (1982). ' Assessing Aspects of Competence in Basic Algebra.' In: *Intelligent Tutoring Systems* (Eds. D.H. Sleeman and J.S. Brown), Academic Press, London. pp 185-199.

Sleeman D. and Brown J.S. (1982). 'Introduction: Intelligent Tutoring Systems.' In: *Intelligent Tutoring Systems* (Eds. D.H. Sleeman and J.S. Brown), Academic Press, London. pp 1-11.

Smith P.R. (1979). 'CAL in Engineering.' In: *Learning Through Computers* (Ed. D.A. Tawney). The MacMillan Press Ltd, London. pp 43-61.

Stanley C.J. (1983). 'Use of Compact Nuclear Operations Simulator for Training Nuclear Personnel.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 2:08.

Stephanopoulos G. (1987). 'Computer Aided Chemical Process Control.' Paper presented at a L.I.S.P.E. meeting at M.I.T. June.

Stephanopoulos G. (1987a). 'Expert Systems and Computing Environments for Process Systems Engineering', Paper presented at a L.I.S.P.E. meeting at M.I.T. June.

Stephanopoulos G. (1990). 'Artificial Intelligence in Process Engineering - Current State and Future Trends.' *Computers Chem. Engng.* Vol. **14**, No. 11. pp 1259-1270.

Tawney D.A. (1979). *Learning Through Computers* (Ed. D.A. Tawney). The MacMillan Press Ltd, London.

Tawney D.A. (1979a). 'An Introduction to CAL.' In: *Learning Through Computers* (Ed. D.A. Tawney). The MacMillan Press Ltd, London. pp 1-10.

Tawney D.A. (1979b). 'CAL and Learning.' *Learning Through Computers* (Ed. D.A. Tawney). The MacMillan Press Ltd, London. pp 109-129.

Thimbleby H. (1985). 'User Interface Design: Generative User Engineering Principles.' In: *Fundamentals of Human-Computer Interaction*, (Ed. A. Monk). Academic Press, London. pp 165-180.

Tompsett P.A. (1987). 'An Integrated Initial Training Programme for a CEGB Operations Engineer.' In: Proceedings of the CSNI Specialist Meeting on Training of Nuclear Reactor Personnel. Orlando Florida, April 21-24 1987. NUREG/CP-0089.

VanLehn K. (1988). 'Student Modeling.' In: *Foundation of Intelligent Tutoring Systems* (Eds. M.C. Polson and J.J. Richardson), Lawrence Erlbaum Associates Publishers, New Jersey, U.S.A. pp 55-78.

Walker E. and Stevens A. (1986). 'Human and Machine Knowledge in Systems.' In: *Intelligent Decision Support in Process Environments* (Ed. E. Hollnagel et al.).NATO ASI Series, Vol. F21. Springer-Verlag Berlin Heidelberg.

Walsh T.E. (1983). 'Adapting United States Training Practices to European Utilities.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 1:04.

Waterfield J.A. and Wyse W.H. (1983). 'Application of User Experience to Nuclear Submarine Training Simulators.' In: *Proceedings of an International Conference Training For Nuclear Power Plant Operation, Bristol. May 1982.* The Institution of Nuclear Engineers, London. ISBN 0 9508960 0 4. Paper No. 4:02.

Wetherill F.E. and Wallsgrove C.S. (1986). 'Operator Training for Startup.' *Plant/Operations Progress* (Vol. 5, No. 2) April 1986. pp 78-80.

Williams G.P. and Hoehing W.W. (1983). 'Causes of Ammonia Plant Shutdowns: Survey IV.' *Chemical Engineering Progress*, March 1983. pp 11-30.

Woolf B.P. (1988). 'Representing complex knowledge in an intelligent machine tutor.' In: *Artificial Intelligence and Human Learning: intelligent computer-assisted instruction* (Ed. J. Self). Chapman and Hall Ltd. New York, U.S.A. pp 3-27.

Zimolong B, Nof S.Y., Eberts R.E. and Salvendy G. (1987). 'On the Limits of Expert Systems and Engineering Models in Process Control.' *Behaviour and Information Technology*. Vol. 6. No. 1. pp 15-36.

Zwaga H.J.G. and Veldkamp M. (1984). 'Evaluation of Integrated Control and Supervision Systems.' In: *Ergonomics Problems in Process Operations.* Institution of Chemical Engineers Symposium Series No. 90. Pergamon Press. pp 133-147.