AN ABSTRACT OF THE DISSERTATION OF

Lukman Irshad for the degree of Doctor of Philosophy inMechanical Engineering and Mechanical Engineering presented on August 06, 2021.

Title: A Framework to Evaluate the Risk of Human- and Component-related Vulnerability Interactions

Abstract approved: _____

H. Onan Demirel Irem Y. Tumer

Most accidents and malfunctions in complex engineered systems are attributed to human error. However, a closer inspection would reveal that such mishaps often emerge as a result of poor design and human- and component-related vulnerabilities acting together. To fully understand and mitigate potential risks, the effects of such interactions between component and human fallibilities (in addition to their independent effects) need to be considered early in the design process. Existing risk assessment methods either quantify the risk of component failures or human errors in isolation or are only applicable during later design stages. This work takes the view that the combined effects of human errors and component failures are better understood when they are studied together. To this effect, this research introduces an early design stage computational framework to model the system level effects of component failures and human errors. Then, an automated fault scenario generation technique and a severity quantification model are introduced to help designers generate a wide range of potential fault scenarios (involving both humans and components) and prioritize them based on severity. Next, the applicability of the framework to complex engineered systems and the accuracy of scenario generation and severity quantification are explored. Finally, this research demonstrates an application of the framework to promote risk-informed ergonomic assessments with the use of digital human modeling simulations. The ultimate goal of this research is to help designers detect

the combined effects of human- and component-related vulnerabilities (in addition to their effects in isolation) in complex engineered systems during early design stages to improve performance and safety while minimizing the potentially costly design changes and rework later in the design stages.

A Framework to Evaluate the Risk of Human- and Component-related Vulnerability Interactions

by

Lukman Irshad

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented August 06, 2021
Commencement June 2022

Doctor of Philosophy dissertation of Lukman Irshad presented on August 06, 2021.

APPROVED:

Co-Major Professor, representing Mechanical Engineering

Co-Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial, and Manufacturing Engineering

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Lukman Irshad, Author

ACKNOWLEDGEMENTS

I have received a great deal of support and guidance throughout this journey. First, I would like to express my gratitude to my advisors Dr. H. Onan Demirel and Dr. Irem Tumer. I am lucky to have had the opportunity to work under your guidance and without your support, this would not have been possible. Dr. Demirel, I am truly thankful for your constant supply of ideas and inspiration. I cannot thank you enough for being constantly available and willing to discuss. Your support and timely feedback have contributed to my growth as a researcher and professional. Dr. Tumer, you have been an excellent mentor. You helped me keep the bigger picture and high-level goals in mind as I progressed through my research. Your contribution to my personal and professional growth is immeasurable.

I would like to thank my committee members Dr. Andy Dong, Dr. Chris Sanchez, and Dr. Judy Liu, whose inputs have improved my dissertation. I am grateful for your time and feedback.

I would also like to express my sincere appreciation to my current and former colleagues at the OSU Design Engineering Lab. Thank you, Salman, Daniel, Hannah, Nico, Katherine, Mihir, Arpan, Kam, and Karina, for your friendship and help. Special thanks and credit to Salman and Daniel for their contribution to my research. Salman helped with the conceptualization of the HEFFR framework and provided expertise in digital human modeling. Daniel helped with formulating the risk quantification model with his expertise in expected cost modeling.

Shout-out to my friends. I truly appreciate your motivation and support throughout this journey. I love you all, and you are family.

Finally, I would like to thank my family, without whom none of this would have been possible. My parents (Irshad and Mazahima), sister (Shimla), brother-in-law (Aslam), and brother (Arshath) have continuously motivated and supported me throughout this journey. I am truly grateful “Momma,” “Dedda,” “Dathi,” “Machan,” and “Arshath” for your sacrifices, unconditional love, and support. Special mention to my nephews and nieces for lighting up my world. You are the best! I would also like to thank my better half Nifla for being patient, supportive, a constant source of inspiration, and

a motivator. You are amazing, and your will, courage, and dedication inspired me to be focused and work harder. I appreciate all that you have done for me. Last but not least, I am grateful to my two precious daughters, Inara and Iqra, for being a constant source of joy and reminding me of what's important in life.

I sincerely thank you all for what you have done for me and your contribution to this journey.

CONTRIBUTION OF AUTHORS

All research in this dissertation was conducted with the guidance of Dr. Irem Tumer and Dr. Onan Demirel. In addition, Dr. Guillaume Brat contributed with the conceptualization of the research presented in sections 3.4. In section 4.5, Dr. David C. Jensen contributed with the conceptualization, revision, and feedback. Salman Ahmed helped with the conceptualization of the fault prediction model of the HEFFR framework in section 3.2. In section 5.5, Salman Ahmed developed the CAD models and provided expertise in digital human modeling. In section 4.5, Daniel Hulse helped develop the fault severity quantification model, produced several figures, and helped write some of the results and discussions. This dissertation was largely compiled from published work, where they were adapted to fit the dissertation structure and revised for flow. All co-authors and publications are acknowledged in the corresponding chapters.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

In dedication to my parents, Mazahima and Irshad, who inspired me to be modest, stay focused, and aim high

Chapter 1: Introduction

This research aims to formulate a computational framework to assess the potential risk of component failures and human errors acting alone and in tandem during early design stages. Also, it explores the use of digital human modeling tools with the risk assessment framework as a means to visualize human-product interactions and inform ergonomic assessments without the necessity for detailed design and physical prototypes. The resulting framework will enable designers to consider human factors starting from early design stages to make better design decisions to minimize system vulnerabilities and improve system performance and safety. In summary, the computational framework introduced in this research can be used to design complex engineered systems that are less likely to have latent or catastrophic failures while minimizing cost and time-to-market.

1.1 Motivation

Let us consider the Boeing 737 Max saga. A malfunction in the angle of attack sensor and the subsequent activation of the Maneuvering Characteristics Augmentation System (MCAS), a flight control software designed to prevent a stall, were identified as the cause of the failure [1]. However, it took two crashes (Lion Air Flight JT610 and Ethiopian Airlines Flight ET302) with 347 fatalities [2, 3] for Boeing to stop blaming human error and admit the design flaw [1]. Subsequently, the Boeing 737 Max was grounded worldwide [4]. The grounding was expected to last a few months until Boeing fixed the problems with the MCAS system. However, it lasted for almost two years because Boeing and FAA kept finding more and more vulnerabilities [4]. The whole debacle is expected to cost Boeing around USD 20 billion in direct costs and more than USD 60 billion in indirect costs [5].

Boeing updated its 737NG to create the 737 Max family [1]. The 737 Max was fixed with a larger engine than the 737NG, causing aerodynamic issues [1]. Boeing tried to solve this hardware issue with

a software fix, the MCAS system [1]. As Captain Sullenberger, who crashlanded an Airbus A320 in the Hudson river put it in his congressional testimony, *"...Though MCAS was intended to enhance aircraft handling, it had the potential to have the opposite effect...it was a trap that was set inadvertently during the aircraft design phase that would turn out to have deadly consequences...the original version of MCAS was fatally flawed and should never have been approved [6]."* He further discussed the nature of failures by adding *"...with older aircraft designs, there were mostly stand-alone devices, in which a fault or failure was limited to a single device that could quickly be determined to be faulty and the fault remain isolated. But with integrated cockpits and data being shared and used by many devices, a single fault or failure can now have rapidly cascading effects through multiple systems...We need to proactively find flaws and risks and mitigate them before they lead to harm....Each aircraft manufacturer must have a comprehensive safety risk assessment system that can review an entire aircraft design holistically, looking for risks, not only singly, but in combination [6],"*

Captain Sullenberger's assessment of the nature of failures in modern aircraft is applicable to any complex engineered system. Like in the Boeing 737 Max crash, human errors are blamed as one of the leading causes of failures in complex engineered systems [7, 8]. Over the past decades, human error related incidents have decreased at a slower rate when compared to incidents attributed to other failures [9]. As a result, around 60%-90% of accidents and performance losses are attributed to human error in aviation, offshore drilling, and nuclear power industries [10–12]. As in the Boeing 737 Max crashes, when mishaps occur, they can be costly and fatal and have lasting effects on the societies, economies, and environment. The partial nuclear meltdown in Three Mile Island, reactor explosion in Chernobyl, and gas leak in Bhopal, India, are all evidence of the heavy toll such human error caused mishaps can have [13]. For example, the Bhopal gas leak caused 3,800 immediate deaths, 600,000 injuries, and another 6,000 casualties since the accident [14]. The surrounding soil and water were found to be contaminated even 20 years since the accident [14]. If one examines these human error caused failures further, it becomes clear that complex interactions between a combination of factors such as human errors, component malfunctions, and poor design trigger these failures [15, 16], where the last link, human (in the form of operators, maintainers, or end users), get blamed.

Hence, to be able to effectively mitigate potential failures, it is important to assess the risk of human and component fallibilities acting in combination during the design process. Design changes made at late design stages are costly and time-consuming [17]. Hence, when potential risks are identified late in the design stage designers are forced to retrofit changes or find workarounds. Often, these changes add new vulnerabilities into the system [18]. For example, in the case of the Boeing 737 Max, the issue caused by the new engine was found later in the design. The MCAS system was built as a workaround to this issue, which introduced new vulnerabilities into the system. Boeing was aware of the vulnerabilities with the MCAS system, and was reluctant to heed to these warnings because of the financial and time pressure [1]. Thus, it is important to identify potential risks early on in the design process so that the potential for design changes later in the design stages can be minimized. However, traditional risk assessment methods assess component failures or human errors in isolation, are only relevant during later design stages, or are applicable to the management of organizations than the design of product interactions. This research aims to overcome these limitations of existing risk assessment methods by formulating an early design stage framework that can assess the risk of electro-mechanical failures and human errors acting in isolation and tandem.

Assessing the risk of human errors and component failures will only allow designers to understand their effects on the system. To fully mitigate risk, one needs to also minimize the ergonomic vulnerabilities embedded within the system because they can negatively affect safety and trigger errors. However, the human interaction and use aspects are often only partially considered or do not receive adequate attention when compared to other product development activities (e.g., operations research, logistics) [19–21]. Traditionally, they are considered later in the design stage with reliance on significant system operational data, which often requires full-scale physical prototypes and extensive human subject data collection [19, 21, 22]. This research aims to overcome this shortcoming by exploring an application of the risk assessment framework by coupling risk assessment with digital human modeling tools to perform ergonomic assessments early in the design process without the need for detailed designs and physical prototypes. In summary, the goal of this research is to formulate a computational framework that can aid designers to mitigate potential failures by allowing them to

better understand the system-level impacts of component failures and human errors acting alone and in combination during early design stages. This research also aims to enable designers to perform ergonomic assessments earlier in design (during early embodiment).

1.2 Research Objectives and Contributions

To realize the above goals, this research pursues three objectives.

- Research Objective 1 aims to formulate an early design stage fault prediction framework that can assess the combined effects of human errors and component failures acting in tandem.
- Research Objective 2 derives a fault scenario generation and prioritization approach to allow designers to identify worst-case fault scenarios based on their perceived severity.
- Research Objective 3 validates the performance of the framework resulting from Research Objectives 1 and 2, and explores an application of the framework to perform risk-informed ergonomic assessments during early design stages.

The following sub sections discuss the expected outcomes, challenges, and contributions of each objective.

1.2.1 Research Objective 1: Assessing the Effects of Human Errors and Component Failures

Research Objective (RO) 1 aims to formulate a computational framework to predict the propagation paths of human errors and component failures during early design stages. To predict the system-level effects of human errors and component failures acting in combination, they both need to be evaluated in parallel. While the parallel evaluation by itself can be a challenge, their reliance on system representations makes the failure prediction even more challenging during early design stages. To accurately facilitate the prediction of the combined effects of human errors and component failures,

the system representation needs to include both product- and human-related data and also capture the relationship between them. Creating a system model with such details can be challenging during the early design stages. Often, detailed models of system components and parameters are not available early in design. Instead, intended system functions are available in the form of functional models. Hence, researchers have used functional models to represent product data during early design stages. However, with the minimal human-product interaction data available during early design stages, representing human-product interactions accurately enough to sustain accurate error prediction remains a challenge.

This research introduces the human aspects of the system to an existing functional model-based fault modeling approach, Functional Failure Identification and Propagation (FFIP) [23, 24], to tackle the above challenges. While FFIP can model the propagation paths of component failures, it cannot capture human error propagation with enough details. This work introduces a graph-based representation of human actions to the overall system representation. A simulation method to model human behaviors in parallel to the component behaviors is developed to predict the propagation paths when fault conditions involving component failures, human errors, or both are injected into the framework. Designers will be able to predict the system-level effects of component failures, human errors, or both using this novel computational fault prediction framework. Since the framework is an early design stage fault prediction tool, designers will be able to use it to understand potential risks early on, and design systems to effectively mitigate potential risks. As a result, the number of costly time-consuming design changes needed later in the design process will be minimized.

1.2.2 Research Objective 2: Identifying Worst-case Fault Scenarios

RO 2 introduces a method to automatically generate potential fault conditions to be assessed using the framework in RO 1 and study how they can be prioritized based on their severity. The framework from RO 1 requires the designers to come up with potential fault scenarios involving both components and humans to assess their system-level effects. Engineers will have to come up with a broad range of

fault scenarios to understand and mitigate potential risks fully. Coming up with such a variety of fault scenarios is challenging or impossible for engineers, which motivates the second RO. Automatically generating fault scenarios that can uncover a majority of potential faults involving humans and components can be a challenge. It becomes even more challenging when those scenarios need to be generated with the minimal information that is present during the early design stages. Even when such scenarios are generated, they need to be successfully prioritized to inform designers about the various levels of risk they can pose. The main challenge in prioritizing fault scenarios early in design is that it needs to be accurate enough to aid informed design decisions while requiring minimal input data. System-designs can change rapidly and multiple candidate designs can be present early in the design process. Therefore, the scenario generation and prioritization setup should be simple enough to allow designers to easily compare and contrast designs, and adapt new system models as potential designs evolve.

To approach these challenges, this research adopts a tree search algorithm to automatically generate a majority of potential critical event scenarios involving both components and humans. Then, a cost and probability model is developed to quantify the relative impact (and thus priority) of critical event scenarios. To calculate the likelihood of the occurrence of critical events, both component failure and human error probabilities are considered, using traditional reliability engineering principles to estimate component failure probabilities and the Human Error Assessment and Reduction Technique (HEART) [25] to estimate human error probabilities. To quantify the relative importance and priority of failures, this research adapts the expected cost of resilience metric developed by Hulse et al. in [26]. Using this approach, designers will be able to identify worst-case scenarios and use failure costs in design trade studies to motivate design. They can also use the resulting information to identify critical points of human intervention and motivate the design of the physical system (e.g., components), electronic system (e.g., control logic and interfaces), and human system (e.g., best practices and training materials).

1.2.3 Research Objective 3: Evaluating the Performance of the Framework

RO 3 aims to evaluate the performance of the framework resulting from RO 1 and 2 by applying it to a complex problem and validating it against real world failures. As with any new fault modeling framework, the framework resulting from this research needs to be validated to understand its capabilities and limitations. Specifically, the framework needs to be analyzed to understand its ability to predict and prioritize failures accurately. If the framework is not capable of generating the worst case fault scenarios or predicting the severity of them with reasonable accuracy, further studies need to be performed to explore the characteristics that make it a useful tool and the areas that need further refinement. The main challenge in performing a comparison study is in choosing the complex engineered system that the framework will be applied on. The system needs to have enough complexity to be able to capture the intricacies of the framework while not being overly complex to a point where modeling becomes infeasible. Another challenge is in identifying real world failures to compare the results against. For the system that is chosen, the historic failure data need to be well documented to make sure that the comparison study is accurate. In summary, a complex engineered system that encompasses enough complexity to allow for realistic modeling while having well documented historic failure data needs to be chosen for this study.

To approach these challenges, a railway locomotive design problem is chosen. The reason for choosing a train locomotive is because it can be modeled in a way that is generic so most railway accidents (regardless of the differences in the train models) can be used in the validation study. The application of the risk assessment framework to complex engineered systems is demonstrated by taking a modular analysis approach. Then, the modular analysis approach is validated against the integral approach to show that the framework can yield consistent results regardless of the analysis being performed modularly or integrally. The results from the train locomotive design is compared against real word train accidents to understand if the framework is able to generate and prioritize the faults that lead to the train accidents accurately. Data for all severe train accidents since the

year 2005 are extracted from the accident investigation databases of National Transportation Safety Board (NTSB) and European Railway Accident Investigation Links (ERAIL) for the comparison study. In summary, this study will help define the effectiveness of the proposed work. Rather than blindly applying the fault modeling method designers will be able to use the proposed work where it appropriately fits and use more caution with regards to its limitations.

In addition to validating the risk assessment framework, this research also demonstrates an application of the framework to inform risk-based ergonomic assessments during early design stages. Complex engineered systems have a large number of potential human-product interactions that require ergonomic assessments. Generally, experts use task analysis to identify the specific ergonomic assessments that need to be performed. This is usually done during late design stages when detailed design data are available. During the early design stages, when minimal product details are available, it might be challenging for experts to identify and prioritize the types of ergonomic assessment needed for a complex engineered system design because of the high uncertainties present. We demonstrate how the risk assessment framework introduced in this research can be used to identify and prioritize the needed ergonomic assessments based on their potential risk. We couple the risk assessment framework with digital human modeling to visualize human product interactions and perform risk-informed ergonomic assessments earlier in the design process without relying on detailed design data and physical prototypes.

On the whole, as shown in fig. 1.1, this research introduces a risk assessment framework that can generate fault scenarios involving humans and components, predict their system-level propagation, and quantify the resulting failures. In addition, a risk-informed digital human modeling based ergonomic assessment approach is explored as an application of the risk assessment framework introduced in this research.

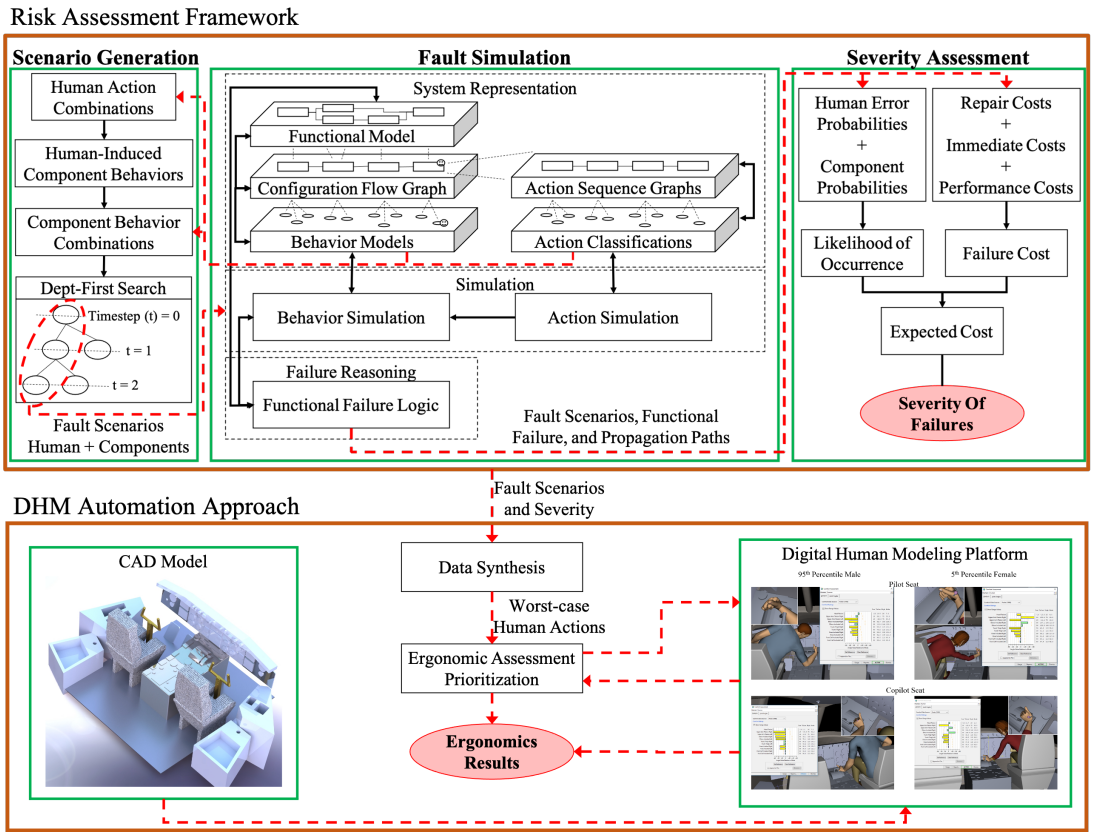


Figure 1.1: Research outcomes: a risk assessment framework to assess human errors and component failures in combination and a risk-informed early design stage ergonomic assessment approach

1.3 Broader Impacts

This project has introduced a framework that not only allows the identification of potential human errors, component failures, and their propagation paths, but also allows designers to identify worse case scenarios, visualize human product interactions, and prioritize ergonomic assessments based on risk, all early in the design process. By making human product interactions digitally available along with the failure data, the framework allows designers to more deeply understand the underlying failure causes and make better-informed decisions early in design. Moreover, the automated scenario generation and prioritization allows designers to perform comprehensive risk assessments that take into account a majority of fault conditions, resulting in systems that are less prone to failures. Also,

by combining the risk assessment relating to humans and components, this framework can act as a merger between two domains (human factors and reliability engineering) within design teams, which will minimize the chances of making trade-offs between the conflicting design decisions made by experts from the different fields.

In addition to the impacts to the field of risk-based design, this research can have an impact on the societies, industries, and environments complex engineered systems operate on. The immediate advantage to society is in the improved performance and safety of products. For the industry, the advantages will be reduced cost and time to market. Furthermore, product recalls due to unforeseen faults can be minimized. In the long term, it can reduce worker injuries and fatalities, reduce the destruction caused to the environment from accidents and performance losses, and minimize asset loss and compensations.

The remaining chapters are organized as follows. Chapter 2 discusses the background that formed the basis for this research. Chapters 3-5 address one research question each. Finally, Chapter 6 summarizes the conclusions of this research and explores future research avenues.

Chapter 2: Background

Traditionally, risk assessments are performed as part of probabilistic risk assessments or safety assessments [27, 28]. Usually, these assessments are performed to understand the human- or machine-related vulnerabilities in isolation by experts from the respective fields. In contrast to this traditional approach, this research takes the view that vulnerabilities in complex engineered systems are better understood when human- and machine-related fallibilities are studied together, making the risk analysis of human-machine interactions a combined effort (by human factors and risk and reliability engineers) rather than an isolated effort. In this chapter, we form the background for this research by exploring existing risk assessment methods. The risk assessment methods are categorized into three: human reliability assessment techniques, component failure assessment techniques, and other risk assessment techniques. The human reliability assessment method are methods that were created to primarily assess human fallibilities. Component failure assessment methods are methods that primarily analyze component failures. The risk assessment methods that do not fall into the human reliability assessment methods and component failure assessment methods categories are detailed under the other risk assessment methods category.

An application of this research demonstrates risk-based digital human modeling approach to enable designers visualize and analyze ergonomic vulnerabilities early in the design stages. This chapter also examines past literature on digital human modeling to study its applicability to design.

2.1 Human Reliability Assessment Methods

Human Reliability Assessment (HRA) is the application of human characteristics and behavior information to design objects, facilities, and environments that require human interactions [29]. HRA is applied in a vast array of domains starting from high-risk industries such as aerospace and

aviation, automobile, and shipping to relatively lower risk industries such as telecommunication, software design, and manual tasks like lathe operation [29]. The goal of HRA is to assess the risk attributed to human error so that human/system vulnerabilities can be reduced to improve safety and reliability [30, 31]. To achieve this goal, HRA relies on identifying human errors, quantifying how likely those errors are prone to occur, and reducing their likelihood of occurrence [30, 32]. As Stanton and Stevenage put it [33], most HRA techniques start with a step by step task break down (hierarchical task analysis, cognitive task analysis, etc.). Then, the potential errors at each step and the psychological error mechanism that causes them are identified. Finally, recovery or error reduction pathways are specified.

Early HRA methods such as Technique for Human Error Rate Prediction (THERP) [34], Human Cognitive Reliability (HCR) [35] and Systematic Human Error Reduction and Prediction Approach (SHERPA) [36] only consider errors of omission and commission giving minimal attention to performance shaping factors (e.g., organizational factors, environmental factors, etc.). On the other hand, methods like Human Error Assessment and Reduction Technique (HEART) [25], Success Likelihood Index Method (SLIM) [37], Standardized Plant Analysis Risk-Human Reliability Analysis (SPAR-H) [38], A Technique for Human Error Analysis (ATHEANA) [39], Cognitive Reliability and Error Analysis Method (CREAM) [40], and MERMOS [41, 42] take performance shaping factors into consideration when analyzing the risk of human errors. Among the above methods, HCR, ATHEANA, CREAM, and MERMOS focus more on the cognitive aspects and the determining factors that lead to an unsafe environment.

Since the inception of the above methods various upgrades and modifications have been proposed. For instance, researches have proposed updates to the HEART method to tailor generic human error probabilities and the performance shaping factors (or error producing conditions) to a variety of industries such as aviation [43], nuclear power [44], railway [45], and maritime [46]. Others have proposed modifications to minimize the subjectivity or reliance on expert judgments in HEART by integrating fuzzy-based methods with HEART [47–49]. Similarly, researchers have also introduced modifications to the SLIM method to minimize the subjectivity that results from the reliance on

expert opinions. For example, Tu, Lin, and Lin [50] proposed combining the SLIM method with Bayesian method to reduce the subjectivity in human error probability calculation [50]. Zhou and Lei [51] integrated the SLIM method with empirical study and complex networks to take into account the interdependence between performance shaping factors when calculating human error probability [51]. Improvements to the CREAM method has looked into tailoring it to specific industries (e.g., aviation [52], maritime [53], spaceflight [54]) and improving the probability calculation model (e.g., better assignment of weights to the performance shaping factors using a Evidential Reasoning (ER) approach [55], minimizing subjectivity using fuzzy-based methods [56]). All of the HRA methods and the modifications that are discussed above require high fidelity task models and detailed component models, narrowing their application to later design stages.

Past research has also introduced HRA methods to be used in early design stages. Examples of early design stage HRA methods include the Technique for Human Error Assessment (THEA) [57], Technique for Early Consideration of Human Reliability (TECHR) [58], and Early Model-based HRA (eMHRA) [59]. THEA uses functional representations instead of component models and usage scenarios instead to task models to enable the early design stage HRA. TECHR was developed with the intention enabling the quantification of human error probability with minimal data requirements. The usage of empherical data, human action models, and human error taxonomies allows this method to be applied during early design stages. The eMHRA method was developed to be used earlier in the PRA in tandem with component failure assessments and emergency operating procedures development. PRA usually requires detailed component models. Hence, this method is more applicable during early late design stages than the actual early design stages.

All of the HRA methods above are static, meaning they only provide insight into a snapshot in time. Recent research has attempted to overcome this limitation by introducing dynamic HRA methods, allowing HRA models to be included into human performance modeling simulations. Some researchers have used simulation and modeling as basis for dynamic HRA. For example, Angelopoulou, Mykoniatis, and Boyapati [60] proposed a HRA method for Industry 4.0 that uses modeling and simulation of dynamic systems to assess human errors when human performs tasks [60]. Other

methods expand upon static HRA methods to enable the dynamic assessment. For instance, Zhang et al. [61] introduced an approach that combined Predicted Mean Vote method and CREAM to analyze mission reliability in dynamic environments [61]. Another CREAM-based dynamic HRA method combines Bayesian network and fuzzy hierarchical task analysis with CREAM to model and quantify human errors in emergency situations [62]. Person Specific Human Error Estimation (PSPHERE) [63] combines the knowledge from existing HRA methods with continuous time Markov chains to assess human error probability in dynamic environments. All of the above dynamic HRA methods require detailed system models and task models, making them only applicable during late design stages. The HRA methods discussed in this section are inadequate in terms of their ability to assess the combined effects of human errors and component failures because they give minimum attention to component failures. In summary, the human reliability methods assess human errors alone. As a result, they are not capable of analyzing the risk of human errors and component failures interacting.

2.2 Component Failure Assessment Methods

Component failure assessment methods aim to analyze the risk of component failures and help designers prevent them. Traditionally, the failure assessments are performed as part of the probabilistic risk assessment of the system under design during later design stages. However, the emergence of risk- and resiliency-based design has resulted in designers performing component failure analysis outside of the probabilistic risk assessment process, especially during early design stages. The ultimate goal of performing component failure assessments is to identify the potential risk of failures early on and design preventive measures into the system so that the potential for catastrophic failures during the use of the system is minimized.

Failure Mode and Effects Analysis (FMEA), Fault Tree Analysis (FTA), and Event Tree Analysis (ETA) are some of the most widely used component failure assessment methods. FMEA [64] systematically decomposes a system into subsystems and then into individual components to determine

failure modes and their effects at a component level and system level. FTA [65], on the other hand, uses a predetermined undesired event to build a graphical model (fault tree) of various parallel and sequential combinations of faults to determine all possible ways that would result in the specified event. Event Tree Analysis (ETA) [66] uses a failure event and creates paths of possible success or failure outcomes to come up with failure propagation paths. In addition, FMEA, FTA, and ETA are used as HRA techniques by human factors experts by switching the context to human errors [29]. However, they are not capable of capturing the combined effects of human errors and component failures acting in combination. Also, these methods are not optimal to be used at the conceptual stage since they require detailed system/component models. They also heavily rely on expert knowledge and historical data, which makes them highly subjective methods.

Researchers have proposed extensions to FMEA, FTA, and ETA with the goal of overcoming some of their limitations. For example, extensions to FMEA aim to minimize the subjectivity and uncertainties in the risk priority number calculations using fuzzy theory [67–69] and multi criteria decision making [70–72]. Extensions to FTA have aimed to manage uncertainty (e.g., using fuzzy theory [73, 74] and enable dynamic modeling (e.g., using Markov chains [75], binary decision diagrams [76], and Petri nets [77]). Similar to FTA, researchers have proposed extensions to ETA with the goal of considering uncertainties (e.g., using fuzzy theory [78, 79] and enabling time-based modeling [80]. Other late design stage component failure assessment methods include bow-tie diagrams and Reliability Block Diagrams (RBD). Bow-tie diagrams [81] combine FTA and ETA to allow the assessment of the causes and consequences of failures. RBD [82], on the other hand, uses a graphical representation of the system where blocks represent components. RBD calculates the system reliability based on how the components are connected (series or parallel).

As an effort to move failure and risk assessment to early design stages, researchers have developed methods that use function-based system models instead of component-based models. Functional Failure Design Method (FFDM) [83], for instance, uses a matrix-based approach to link system functions to potential failures before any component selections are made. Lough, Stone, and Tumer [84] expanded FFDM to quantify the likelihood of failures and risk with a goal of giving designers a

quantitative measure of potential functional failures and their effects during early design stages [84]. Conceptual Stress and Conceptual Strength Interference Theory (CSCSIT) [85] was developed with the goal of enabling the use of Stress and Strength Interference Theory for reliability assessment during early design stages. It uses conceptual strength coefficients, conceptual stress coefficients, and conceptual failure analysis to come up with failure probability of function failures.

Functional Failure Identification and Propagation (FFIP), Conceptual Object-Based Risk Analysis (COBRA), and the function-based failure propagation method go a step further and allow the designers to analyze the downstream effects of failures. FFIP [23] uses a graph-based system representation that includes a system functions, generic components, and behavior models to analyze the functional failures and their propagation paths. COBRA [86] converts the functional representation of the systems to a mathematical model and identifies potential failures and failure flow paths. The function-based failure propagation method [87], on the other hand, analyzes chains of functions to provide the likelihood of a failure propagating to a function and the possibility of any function propagating a failure based on historical failure data. Recent research has explored the use of Bayesian networks for early design stage failure assessments. For example, Goswami and Tiwari [88] proposed a risk assessment methods for modular complex systems where both technical and commercial risk parameters are represented through parent and root nodes in a Bayesian network. Yodo and Wang [89] proposed a method where potential failures are considered as internal and external disruptions and represented as root nodes in the Bayesian network.

With complex engineered systems becoming more and more software intensive, recent research has looked into risk assessment methods that assess failures in such software driven complex engineered systems. Researchers have used FFIP as a tool to model failures in such software driven complex engineered systems [90–92]. Others have used Markov chains [93, 94] and environmental modeling [95–97] based methods to assess such failures. These methods are capable of assessing hardware, software, and hardware-software interaction failures. In brief, the component failure assessment methods either require detailed models and hence, are only applicable during later design stages or give minimal attention to human elements of the system.

2.3 Other Risk Assessment Methods

There are risk assessment methods that do not fall into the category of component failure assessment methods or human reliability assessment methods. For example, systemic risk assessment techniques look into system-level vulnerabilities in sociotechnical systems rather than looking into human error or component failures specifically. These techniques also consider organizational vulnerabilities and their effects. Systems Theoretic Accident Model and Process (STAMP) [98], Functional Resonance Analysis Method (FRAM) [99], and modified Event Analysis of Systemic Teamwork (EAST) [100] are examples of systemic risk assessment methods. Originally, STAMP and FRAM were developed as accident analysis techniques. Experts have used them as risk assessment tools during system design (e.g., STAMP [101–103], FRAM [104–106]). STAMP [98] uses system and control theory to identify potential system vulnerabilities where failures are seen as an emergent property of the systems. Specifically, it views accidents as violations of safety constraints relating to behaviors FRAM [99], on the other hand, is a function representation based method that uses combinations of normal performance variability to identify potential vulnerabilities. Stanton and Harvey [100] proposed a modification to the EAST method to enable the use of it to assess the risk of sociotechnical systems [100]. This approach represents the system as networks of networks and assesses failures in information commutation through broken-links. These methods are based on a high-level organizational system model; thus, are more applicable to the organizational level rather than the design of human product interactions.

More recently, researchers have introduced methods that are meant to capture human-product interaction related failures during early design stages. One such method combines the results from FFDM and SHERPA to understand both component failures and human error[107]. Function Human Error Design Method (FHEDM)[108] is inspired from FFDM. It uses a series of matrix multiplications to identify the potential human errors for each function in the functional model. The overall goal of these methods is to provide designers insight into potential human errors and component failures so that mitigating actions can be taken early in design before any costly design commitments are made. While these methods allow designers to assess component failures and human errors together,

they do not assess their propagation. As a result, they fall short in terms of their ability to capture the interaction effects of component and human vulnerabilities. The risk assessment framework introduced in this research differs from the human-product interaction failure assessment methods discussed above in its ability to predict the system level effects of human errors and component failures acting in combination during early design stages.

In summary, the risk and reliability assessment methods discussed in this chapter either assess component failures or human errors alone, are only applicable during late design stages, or do not assess the propagation of human errors and component failures acting together. This research aims to overcome these limitations by introducing a early design stage risk assessment method that can assess and quantify how human errors and component failures affect the overall system health when they act in combination (or alone) so that designers can consider mitigation strategies before any significant design commitments are made.

2.4 Digital Human Modeling

This research utilizes Digital Human Modeling (DHM) simulations to visualize human-product interactions and perform risk-informed ergonomic assessments. Formally, DHM is used to digitally represent and control a human to visualize human-system interactions in a computer and apply human factors principles to improve safety and performance [19, 109, 110]. DHM allows engineers to apply human factors principles proactively to evaluate human and system performance before any physical prototypes are made [109, 111]. This approach reduces the need for human subject data collection in full-scale physical prototypes (which are costly and time-consuming) in empirical human performance studies.

The application of DHM was confined to complex engineered systems and large-scale project domains such as military and space in its early days. However, in recent times, its application has broadened to a variety of industries including aviation, automobile, healthcare, manufacturing planning, assembly planning, and workspace planning. For instance, in the automobile industry, DHM

is used to assess how the steering wheel location affects the visibility of the instrument clusters [112]. Additionally, the reach volume for the primary controls and vision obscuration zones are analyzed using DHM [112]. One application of DHM in the healthcare industry looks to understand the influence of muscle weakness, decreased range of motion, and pain on the functional abilities of the population that uses ambulatory aids such as wheelchairs, canes, and walkers [113]. In the assembly and manufacturing industries, DHM is often used to evaluate workers' posture and comfort. For example, a study of workers who use a riveting system, aimed to avoid potential injuries caused by lifting heavy weights, taking awkward postures, and performing repetitive motions for long periods by assessing worker comfort and identifying ideal workspace configurations that are suitable for a wide range of populations [112]. Another study used DHM in an aviation-related design project to assess pilot performance during emergency procedures. The study found that DHM, coupled with a motion capture system, can help with successfully identifying individual postural strategies without the need for excessive physical prototyping [114]. In summary, DHM is used to assess physical ergonomics relating to reach, vision, posture, comfort, and many more in a variety of industries.

While DHM does a satisfactory job in representing the physical aspects of a human, its ability to capture cognitive aspects of humans is still lacking. There are a few tools that are still in research focusing on integrating cognitive ergonomics with DHM. These cognitive models primarily focus on the perceptual-cognitive aspects of human performance. Since cognitive elements of a human can be abstract when compared to physical elements simulating them can be very complicated. In addition, human perception and cognition are complex processes. These and the variations in individuals and populations has meant that cognitive ergonomics are not as well developed as physical ergonomics in DHM platforms [115]. However, the tools that are available within DHM platforms are still handy when it comes to visualizing human product interactions and aiding design decision making relating to the physical elements of a human.

Chapter 3: Assessing the Effects of Human Errors and Component Failures

This chapter addresses research objective one by formulating and validating a fault prediction framework to assess the effects of human errors and component failures acting in combination and isolation. In contrast to traditional approaches that either study human errors or component failures in isolation, the framework introduced in this chapter approaches the fault prediction with a goal of identifying the system-level propagation of component and human fallibilities interacting and acting alone. This research was published in the ASME Journal of Computers and Information Science in Engineering and in the Proceedings of the 2018 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference and was coauthored by Lukman Irshad, Salman Ahmed, H. Onan Demirel, and Irem Y. Tumer [116, 117]. Next, the fault prediction model of the framework is validated using the Air France 447 crash and a high-fidelity flight simulator to confirm that the predicted faults were realistic representations of real world failures. This research was published in the ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems Part B: Mechanical Engineering and was coauthored by Lukman Irshad, H. Onan Demirel, Irem Y. Tumer, and Guillaume Brat [118].

3.1 Motivation

As discussed in Chapter 1, when failures occur in complex engineered systems, often times, they are caused by complex interactions between poor design, component malfunctions, and human fallibilities [16, 107]. Designers need to be able to foresee such complex interactions to be able to effectively mitigate the risk of potential failures in complex engineered systems. However, with such complex systems, modeling all possible failure scenarios and associated risk is beyond the grasp of

any designer, and handling critical events, abnormal situations, and deviations safely and effectively become almost impossible for operators without proper operational procedures and training [23]. This chapter tackles this issue by proposing ways to help designers understand potential failure scenarios, human errors, and their effects over time to the overall system.

One way to address the above issue is to incorporate human factors into the design process. However, the human interaction and use aspects often are only partially considered or do not receive adequate attention when compared to other product development activities (e.g., operations research, logistics) [19–21]. Traditionally, they are considered later in the design stage with reliance on significant system data, which often requires full-scale physical prototypes and extensive human subject data collection [19, 21, 22]. Design changes made during the later design stages (after the design has been established or during prototype testing) can be very costly and time consuming compared to design changes made during early design [17]. This forces engineers to find workarounds or retrofit changes when system vulnerabilities are found during later design stages. Hence, the objective of this research it to not only identify potential failures that result from interactions between component and human vulnerabilities but also identify such potential risks early in design so that the overall cost and time to market are reduced. Often, detailed models of system components and parameters are not available early in design. Instead, intended system functions are available in the form of functional models. The above limitations prompt us to explore an approach that utilizes the functional representation of systems during early design.

Existing risk assessment methods and tools fail to cover all aspects of the question formulated above. Depending on the method, the existing tools either require detailed models of system components, rely mostly on expert knowledge, address human error at an abstract level, analyze only human errors, or fail to provide insight on how the failures will propagate and affect the system overall. This work expands Functional Failure Identification and Propagation (FFIP) [23], a functional model-based failure prediction framework, to include human aspects of the system to the failure prediction. During the early design stages, there are minimal details about the tasks the users need to perform, making it hard to perform a task analysis to fully capture human-product

interactions in a way that it promotes error prediction. Thus, this work takes a novel approach to model human-product interactions through Action Sequence Graphs (ASG), where the focus will be shifted from the use of traditional task analysis approaches to human action based interaction representations. Here, human-product interactions are represented using the actions that a human will perform to interact with a component (i.e., reach, grasp, and turn valve) rather than being represented using the tasks (i.e., reduce flow).

The framework introduced in this research defines the human behavior within the context of the system operation through action classifications by defining all possible nominal and faulty action states for each action in the ASGs. An action simulation algorithm takes fault scenario inputs involving humans to predict the resulting human-induced behaviors of components using the action classifications and ASGs. With the added modules, the expanded framework will be able to take fault scenario inputs relating to humans and components, perform a time-based fault modeling simulation, and output human errors, functional failures, and their propagation paths. In the second half of this chapter, the proposed framework is validated to evaluate if the failures it predicts are realistic by modeling the Air France 447 crash using the framework and comparing the results with what happened in reality. Additionally, a comparison is performed between results from randomly derived fault scenarios modeled using the proposed framework and a high-fidelity flight simulator. Since the capabilities of FFIP has been well studied in previous work [23, 24, 92, 119, 120], the validation only examines the new modules introduced in this research. The results show that the proposed framework can predict potential failures with reasonable accuracy. However, it lacks fidelity when compared to real-world events and simulator results.

Overall, in addition to allowing the designers to analyze functional failures and their propagation paths at a functional level [24], the proposed method will help designers identify potential human errors and understand how they are produced at an early design stage before any potentially costly design decisions are made. With the resulting data, a designer may choose to apply necessary human factors guidelines or suggest operating procedures or training to mitigate human errors. Similarly, a designer will have the ability to choose appropriate components and functions that

mitigate component failures [23, 24, 92, 119, 120]. The overall capability of the proposed method will not only prevent performance losses, failures, and accidents but also reduce cost and time to market of complex engineered systems.

3.2 Background

In this section, the Functional Failure Identification and Propagation (FFIP) Framework is discussed in detail as it is used as the basis for the framework developed in this research.

3.2.1 Functional Failure Identification and Propagation Framework

FFIP, introduced by Kurtoglu and Tumer [23, 24], is capable of identifying functional failures and their propagation paths during the conceptual design stage. The FFIP framework includes a graphical system model, a behavioral simulation, a reasoning logic called Functional Failure Logic (FFL).

Three graphical representations of the system are used: Functional Model, Configuration Flow Graph, and Behavior Model. The functional modeling is done using the Functional Basis for Engineering Design (FBED) method [121], where the overall desired/actual function of the system is decomposed into smaller sub-functions and flows using a standard taxonomy. For each function or set of functions, generic components are then chosen. The Configuration Flow Graph (CFG) is then built such that the nodes represent the components, and the arcs represent the flow of energy, material, or signal between the components. The flows are named using the taxonomy from FBED, and the components are named using a standard taxonomy for electro-mechanical components. Finally, the behavior of the system is represented using a qualitative model where each nominal and faulty discrete modes are derived using the input-output relations between each node in the configuration flow diagram. For example, the component “pipe” may have three discrete behaviors depending on the input-output relations of the liquid flow.

- Nominal: Inflow equals outflow.

- Failed Ruptured: Outflow equals zero.
- Failed Leak: Outflow less than inflow.

The behavior simulation uses the behavior model and the CFG to evaluate the evolution of the overall system state for different input scenarios. Finally, FFL uses the state changes resulting from the behavior simulation to classify each function as operating, degraded, or lost. The functional failures and their propagation paths are then produced for each input scenario. Even though FFIP allows the detection of potential system failures and their propagation paths using critical event scenario and human error inputs at the conceptual design stage, it fails to capture the human actions that contribute towards producing the human error, giving designers minimal insight into mitigating human errors.

3.3 A Framework to Model Human Errors and Component Failures in Combination

The purpose of this research is to establish a formal method to be used in conceptual design to identify functional failures and possible human errors in complex engineered systems and simulate how they propagate to affect the system. FFIP uses function, structure, and behavior modeling to simulate failure propagation paths and resulting functional failures. We introduce the Human Error and Functional Failure Reasoning (HEFFR) framework, which captures human errors and their propagation paths in addition to functional failures and their propagation paths [23, 24] by integrating Action Sequence Graphs, Action Classifications, and Action Simulation into the FFIP framework. The architecture of HEFFR is illustrated in Fig. 3.1.

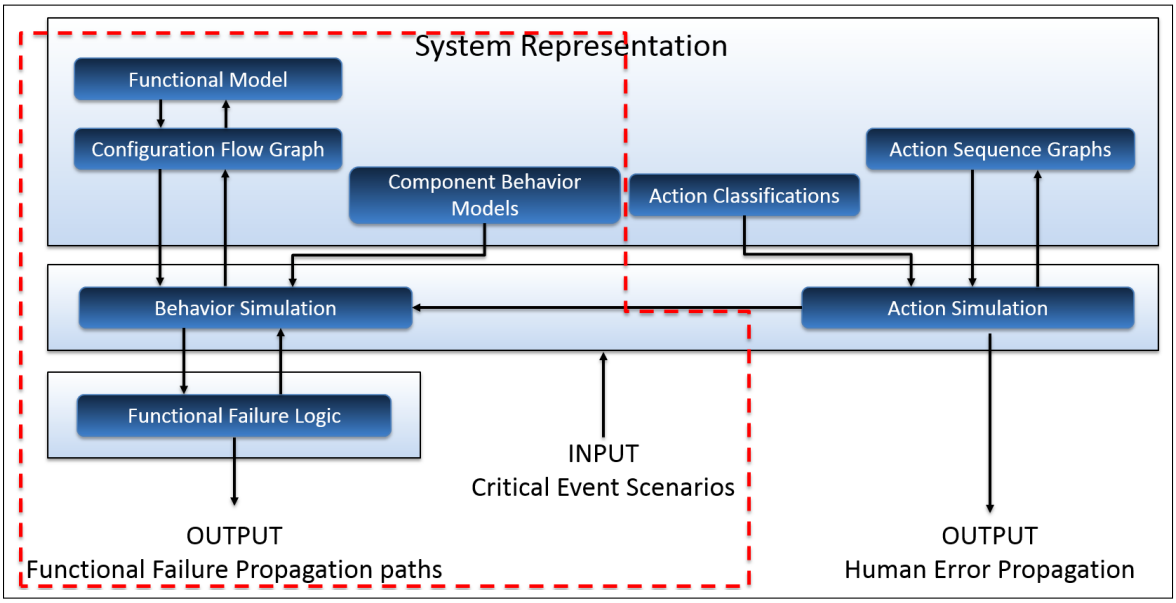


Figure 3.1: The architecture of Human Error and Functional Failure Reasoning (HEFFR) framework (area highlighted by dotted lines indicate modules from FFIP)

3.3.1 Human Error and Functional Failure Reasoning Framework

Note that this section does not go into the details of creating functional models, configuration flow graphs, and behavior models. In addition, the behavior simulation process and the functional failure logic are not explained. Since these modules are fairly well explained in the literature related to FFIP [23, 24, 92, 119, 120], this section focuses mainly on the modules that are new to the HEFFR framework.

3.3.1.1 Action Sequence Graphs

The accuracy of how the human errors and their propagation are determined is highly dependent on how human-system interactions are represented. A model that is capable of capturing human-system interactions in a way where all possible human actions are represented as a flow will facilitate both error recognition and propagation. During conceptual design, the extent of human-system interactions

is only known at an abstract level. Traditional task analysis methods require significant human and system information to represent human-system interactions. Hence, using task analysis to represent human-system interactions during early design stages will not promote accurate fault prediction. Action Function Diagrams [122, 123], on the other hand, facilitate the determination of human-system interaction using functional models. However, they do not capture all possible human actions as a flow. Hence, this research introduces a novel graph-based human-product interaction representation model called Action Sequence Graphs (ASG) to represent the human-product interactions as action sequences.

Representing the human-product interactions through actions (i.e., reach, grasp, and turn valve) rather than through tasks (i.e., reduce flow) comes with several advantages. First, it will allow an easy-to-construct human-product interaction representation with the minimal data available during early design stages. Since ASGs tie human actions to components, a direct link between human interactions and components will be present, making the prediction of combined effects of human errors and component failures more plausible. Finally, multiple users can be represented by creating multiple ASGs for the same component, making the error prediction even more realistic.

Action Sequence Graphs (ASG) are created for all components that require human interaction. ASG is a graphical representation of the action sequence that needs to be performed by the human to interact with the component. Each action in the sequence takes outcomes from the other actions or stimulus as inputs and produces outcomes as outputs. For example, if an operator has to turn a valve off, first he or she needs to Look at it. Then, Reach, Grasp, and finally Turn the valve. Here, the action Reach takes the outcome from the action Look as an input and the output produced by it acts as an input to the action Grasp. Figure 3.2 shows a generic action sequence graph where Stimulus (S) 1 acts as an input to Action 1 to produce Outcome (O) 1. Then, O1 and S2 become the input for Action 2 to produce O2. Here, the outcome for Action 2 results in the direct manipulation of the system while all other actions indirectly contributed towards it.

There are two general heuristics that need to be followed when constructing ASGs. The actions represented in the ASG should be at the lowest possible level. In other words, the actions should

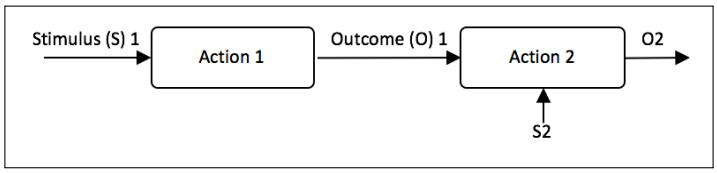


Figure 3.2: Generic Action Sequence Graph

not have any further breakdowns. For example, the action Press (representing pressing a button) can be broken down further into Look, Reach, and Push. Hence, Press should not be included in the ASG. Instead Look, Reach, and Push should be included. Secondly, ASG should include all possible actions that will contribute towards interacting with the component regardless of them being required or not. These heuristics make sure that all possible actions are analyzed which in turn will increase the accuracy of the human errors and their propagation paths produced by HEFFR.

3.3.1.2 Action Classification

The next step is to classify all nominal and faulty outcomes for each action in the Action Sequence Graphs. For example, the action Reach can have four possible classifications;

- Reached - nominal: Reached the expected object.
- Reached - failed: Reached an unexpected object.
- Cannot reach: Cannot reach the expected object.
- No action: No action was taken.

Often, the action classifications for the same action are re-usable across the system, since their nominal and faulty outcomes do not change. For instance, the action Reach in the ASG for a lever and valve will have the same classifications because the faulty and nominal outcomes for the action Reach is the same in both cases.

3.3.1.3 Action Simulation

Action simulation is performed to understand how different human actions will impact the interaction between the human and the specific component. The simulation happens in two steps. During step 1, action simulation takes the critical event scenarios as inputs and simulates the human activity using the ASG to produce action classifications. Here, the action classifications will be a function of stimulus, outcomes from the previous action, and the outcome of the action simulated. The outcomes for each action will come from what the human user did, representing a discrete state such as no action, nominal, or faulty. For instance, for the action Grasp, if the human grasps the object, the outcome will be "grasped," and if he or she fails to grasp, the outcome will be "not grasped." If the action was not attempted, the outcome will be "no action." Depending on the type of outcome from the previous step, the status of the stimulus, and/or the outcome of the action simulated at a given time step, the action classification for that time step will be assigned for each action.

In step 2, the evolution of each action classification is traced by the use of ASG. The ASG provides a graph-based formal representation of individual actions as a flow to be integrated to produce human-component interactions. Accordingly, action classifications of each action at a given time are analyzed to come up with the behavior state of the components that require human interaction. Here, the human errors and how they propagate to affect the behavior of system components are tracked. Finally, the resulting behavior state at each time step and the behavior models for the other components in the system are fed into behavior simulation, and subsequently into FFL to identify functional failures and their propagation paths.

The critical event scenarios are potential failure event scenarios that include both human errors and component failures. They may represent human errors and components failures individually or collectively. The designers are encouraged to devise critical event scenarios that could potentially cover all potential failure conditions and use scenarios as comprehensively as possible.

In summary, HEFFR uses a functional model, configuration flow graph, component behavior models, and action sequence graphs to represent the system and the human-system interactions. It takes critical event scenarios as inputs to produce potential human errors, functional failures and

the propagation paths as output. CFG and behavior models are used in the behavior simulation to determine non-human induced behavior modes of components. ASGs and action classifications are used in action simulation to determine human induced behavior modes of components. Action simulation also identifies human errors and how they propagate to affect the behavior of system components. Finally, the resulting behavior model is used in FFL to identify functional failures and their propagation paths. Similar to FFIP, the simulation operates by solving a timed simulation in the intervals between discrete events. When an event occurs, the simulation is stopped, and the corresponding mode transition is executed. Stopping the simulation allows the input of critical events/scenarios at any given time step. The simulation may run for a predetermined number of time steps or until the system reaches a specific end state.

3.3.1.4 Validation

After implementing HEFFR, designers may choose to add redundant functions, modify design specifications, modify component choices, perform additional analysis (e.g., Finite Element Analysis (FEA) for structural integrity), or suggest additional testing to mitigate function/component related failures. They may also apply HFE principles to derive design specifications, suggest operational procedures, or recommend training to mitigate human errors. We recommend coupling Digital Human Modeling (DHM) with HEFFR as a means of applying HFE principles for non-cognitive human actions [15, 19]. DHM based human-machine simulations allow for a visual representation of the human-system interactions at an early design stage. HEFFR does not consider ergonomics when implemented by itself. However, when coupled with DHM, ergonomic analysis can be performed in the DHM environment [15, 19] and the results can be used to modify the HEFFR system model. This permits the inclusion of ergonomics-related design parameters within the HEFFR framework. The coupled use of DHM and HEFFR not only provides a means to represent human-system interactions visually, but also allows for non-cognitive human factors analyses such as reach, vision, and comfort [15, 19]. It also acts as a means to apply HFE principles to mitigate potential human errors starting from

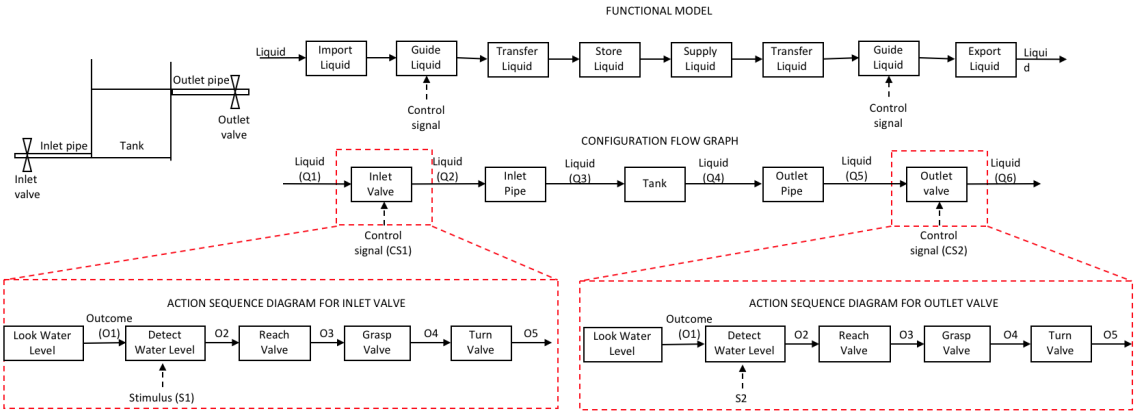


Figure 3.3: System model of a hold up tank

early design stages. The application of HFE principles early in design will reduce the necessity for major design changes at later design stages and hence bring down cost and time to market.

3.3.2 Example: Hold-Up Tank

The capabilities of FFIP to identify functional failures and their propagation paths are fairly well tested and documented [23, 24, 92, 119, 120]. Since HEFFR uses the same mechanism as FFIP to identify functional failures and their propagation, we only evaluate human errors and their propagation in this research. The evaluation of potential human errors and their propagation of a liquid tank concept is presented. Different versions of this tank problem have been widely studied by various researchers [23, 124–127]. This problem was chosen by Kurtoglu and Tumer to initially evaluate the FFIP framework [23].

The problem is to design a system that is capable of regulating the amount of liquid in an open tank. To implement HEFFR, the system model is generated by creating the functional model, CFG, and ASG. Figure 3.3 shows a schematic of the system model. The overall system will have two pipes, two valves, and a tank. Both inlet and outlet valves are controlled manually by an operator. Aircraft pilots and submarine operators may face situations where they have to monitor gauges or instrument panels, and depending on the reading, are required to apply certain controls manually [128–130].

Table 3.1: Action classifications for a valve

Actions	Nominal and Faulty Responses			
Look	Visible	Not Visible		
Detect	Detected - Nominal	Not Detected -Nominal	Detected – Failed	Not Detected - Failed
Reach	Reached – Nominal	Reached - Failed	Cannot Reach	No Action
Grasp	Grasped	Cannot Grasp	No Action	
Turn	Turn to Close	Turn to Open	Cannot Turn	No Action

Table 3.2: Behavior modes of a valve

	Behavior Modes			
Human Induced	Nominal Off	Nominal On	Failed Close	Failed Open
Non-Human Induced	Stuck Open	Stuck Closed		

Table 3.3: Possible outcomes for actions performed on a valve

Actions	Possible Outcomes			
Look	Clearly Visible	Barely Visible	Not Visible	
Detect	Detected	Not Detected	Not Attempted	
Reach	Reached	Reached - False	Not Reached	Not Attempted
Grasp	Gasped	Not Grasped	No Attempted	
Turn	Turned Clockwise	Turned Counter-clockwise	Not Turned	Not Attempted

The manual nature of the valves was chosen as a simplified version of such scenarios. The operator is expected to shut off the inlet valve if the water level reaches a specific overflow threshold. Similarly, they are expected to shut off the outlet valve if the water level reaches a certain dry-out threshold. Two assumptions are made to simplify the problem: The flow is uninterrupted and the operator is continuously monitoring the system without any breaks. Overall, there are two components that require human interaction, 5 actions each per component, and 17 action classifications per component. Tables 3.1 and 3.2 show the action classification for the outlet valve and the behavior modes of components inlet valve and outlet valve. Table 3.3 shows the possible outcomes for each action. Here, the valve is assumed to shut off when turned counter-clockwise and turn open when turned clockwise. In addition, Figs. 3.4 and 3.5 show the action simulation logic step 1 for actions Detect and Reach and action simulation step 2 for the outlet valve.

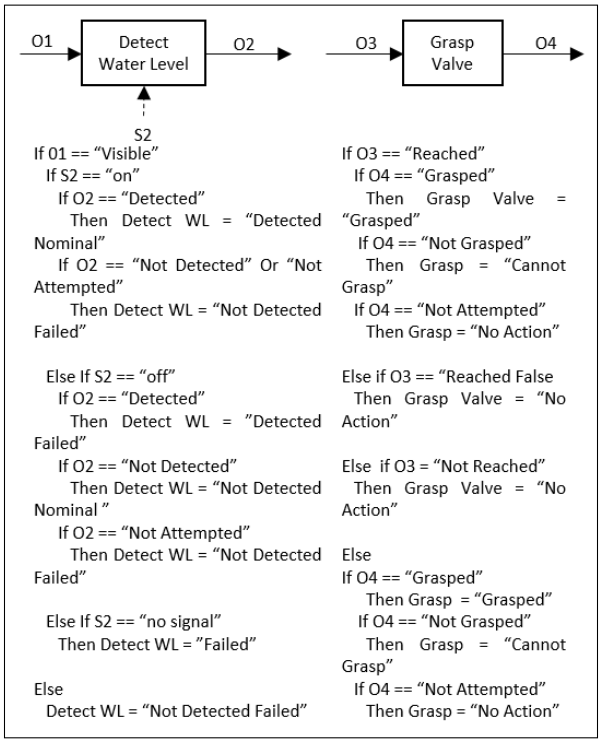


Figure 3.4: Action simulation step 1 for actions Reach and Grasp

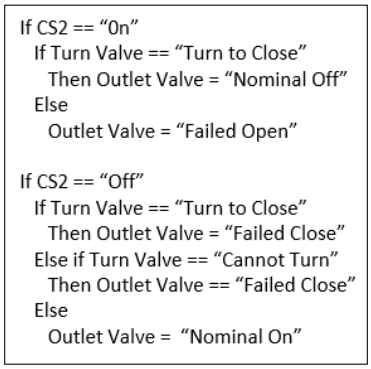


Figure 3.5: Action simulation step 2 for outlet valve

The system is analyzed under two scenarios. Through this analysis, HEFFR will answer questions such as "what happens if the operator cannot reach the valve?", "what will the impact be if there were two operators involved and a miscommunication occurred?" and "what are the effects of closing

the wrong valve?." Then, using the results from HEFFR, DHM is used to analyze the critical non-cognitive actions to derive design specifications. The following section goes into details of the three scenarios and the results of the HEFFR and DHM analyses.

3.3.3 Results

Three event scenarios involving human-system interactions are formulated and implemented as an initial application of HEFFR. To start the simulation, the behavior modes of all components are set to nominal. Then, the action classifications and the input state variables for each component are initialized. The state of both CS1 and CS2 (refer Fig. 3.3) are "off" at the initial stage (time = 0). The initialization of the action classifications is done to reflect this state. For example, for the outlet valve, Look equals "visible," Detect equals "not detected - nominal," and all other actions equals "no action."

The first scenario includes the following events. The water level goes below minimum, the operator detects it and turns off the wrong valve (inlet valve instead of outlet valve). It is assumed that the operator detects the low water level and follows up with the actions in the same time step. While delays in detection and follow up actions can be modeled and the resulting system states can be inferred, the scenarios in this case study purposefully omit delays so that the focus can solely be directed towards the new modules of HEFFR. Screenshots of the action responses, action classifications and the resulting behavior modes of both inlet and outlet valve at time steps(t) 0, 5, and 6 are shown in Fig. 3.6.

At t = 5, the events described above are fed into the simulation. The state variable for CS2 is set to "on." For the outlet valve, the outcome of the actions Look, Detect, and Reach is set to "clearly visible," "detected," and "reached - false" respectively. Here, the outcome "reached - false" means that the wrong object was reached. For the inlet valve, the outcome of actions Reach, Grasp, and Turn is set to "reached," "grasped," and "turned counter-clockwise." Using these inputs, the action simulation algorithm determines the action classifications for each action in the ASGs. Then, using

Time Steps		**t = 0**	**t = 5**	**t = 6**
Contro Signal 1		off	off	off
Contro Signal 2		off	*on*	*on*
Action Outcomes				
Components	**Actions**			
Inlet valve	Look	Clearly Visible	Clearly Visible	Clearly Visible
	Detect	Not detected	Not detected	Not detected
	Reach	Not Attempted	Not Attempted	Not Attempted
	Grasp	Not Attempted	*Grasped*	*Grasped*
	Turn	Not Attempted	*Turned Counter-clockwise*	*Turned Counter-clockwise*
Outlet valve	Look	Clearly Visible	Clearly Visible	Clearly Visible
	Detect	Not detected	*Detected*	*Detected*
	Reach	Not Attempted	*Reached - False*	*Reached - False*
	Grasp	Not Attempted	Not Attempted	Not Attempted
	Turn	Not Attempted	Not Attempted	Not Attempted
Action Classifications				
Intel Valve	Look	Visible	Visible	Visible
	Detect	Not Detected - Nominal	Not Detected - Nominal	Not Detected - Nominal
	Reach	No Action	No Action	*Reached - Nominal*
	Grasp	No Action	*Grasped*	*Grasped*
	Turn	No Action	*Turned to Close*	*Turned to Open*
Outlet Valve	Look	Visible	Visible	Visible
	Detect	Not Detected - Nominal	*Detected - Nominal*	*Detected - Nominal*
	Reach	No Action	***Reached - Failed***	*No Action*
	Grasp	No Action	No Action	No Action
	Turn	No Action	No Action	No Action
Component Behavior States				
Inlet Valve		Nominal On	**Failed Close**	**Failed Close**
Outlet Valve		Nominal On	**Failed Open**	**Failed Open**

Figure 3.6: HEFFR simulation scenario 1

the action classifications, the behavior state of the inlet valve and outlet valve is determined as “failed close” and “failed open,” respectively. This information is fed into the behavior simulation and then into FFL to determine the failure propagation. Since the inflow is shut off and the outflow is still on, the water will continue to flow out and result in a tank dry out. The human error of choosing

the wrong valve propagates to the loss of function Guide Liquid immediately after the non-nominal behavior of the components is observed. Then the function Store Liquid is lost due to the dry out.

The second scenario involves two operators, where Operator 1 performs the same actions as scenario one but fails to communicate his or her detection and follow-up actions with Operator 2. Later, Operator 2 (before a tank dry out) notices the low water level and closed the inlet valve, and turns the inlet valve back on. However, thinking that he or she has done enough to prevent a dry out, the operator fails to follow the operating procedure and turn off the outlet valve. The simulation is initiated and conducted similar to scenario 1 for parts involving operator 1 (until t = 10). Then, at t = 10, the events pertaining to Operator 2 is inserted into the simulation. This leads to the behavior mode of the inlet valve to go back to "nominal on," and the function Guide Liquid relating to the inlet valve is restored. However, since she failed to turn off the outlet valve, it continues to operate non-nominally. Even though the action taken by Operator 2 improved the state of the system, it continued to operate with functional losses. The continuous presence of the failure eventually leads to a tank dry out and the function Store Liquid is lost. The failed action by Operator 1 causes the system to start operating at a failed state. Operator 2 may have improved the system state. However, their failure to follow the procedure and turn off the outlet valve resulted in the system operating non-nominally and eventually completely losing function.

The results from the scenarios analyzed above show that the actions Detect and Reach play an important role in keeping the system failure-free. In this example, the valves should be placed within the reach of a 5^{th} percentile U.S. female to make sure that they can be accessed by a majority of the U.S. population. A reach analysis is done using DHM to make sure that the valve is within the reach of a 5^{th} percentile U.S. female. Figure 3.7 shows the 5^{th} percentile U.S. female manikin from the Anthropometric Survey of U.S. Army Personnel (ANSUR) anthropometric library and the reach envelope. Unlike Reach, the action Detect is purely cognitive. Since DHM tools do not include cognitive tools to assess psychological parameters, designers will have to rely on human subject data collection and subjective multidimensional data collection methods to measure cognitive performance. Instead of relying purely on visually inspecting the tank, the designers may choose to add an alarm

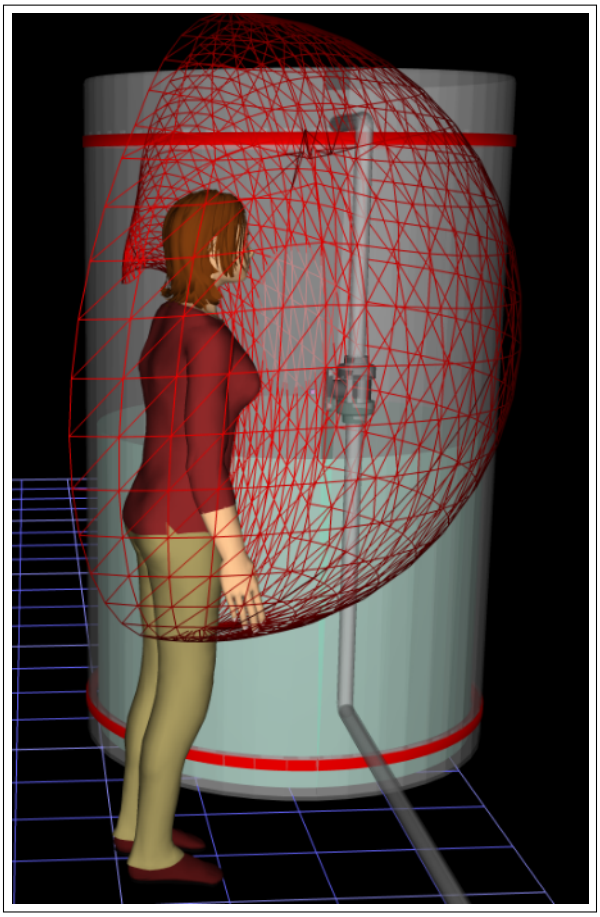


Figure 3.7: Reach envelope of a 5^{th} percentile U.S. female

to notify the operator when the water level is too low or too high or they can suggest training to make sure that the operators detect the water levels as expected.

Our simplified example does not include many of the actual elements of operation or control rooms. An actual work environment might require elements such as, tables, monitors, computers, etc. For example, with the addition of a chair and table, the action sequence of the operator is different. Also, DHM analyses considering percent vision obscuration and reach need to be performed to make sure that the new elements of the operation or work environment do not obstruct the vision or the reach of the operator. DHM is also used to determine the action sequence (Look, Detect, Stand-up,

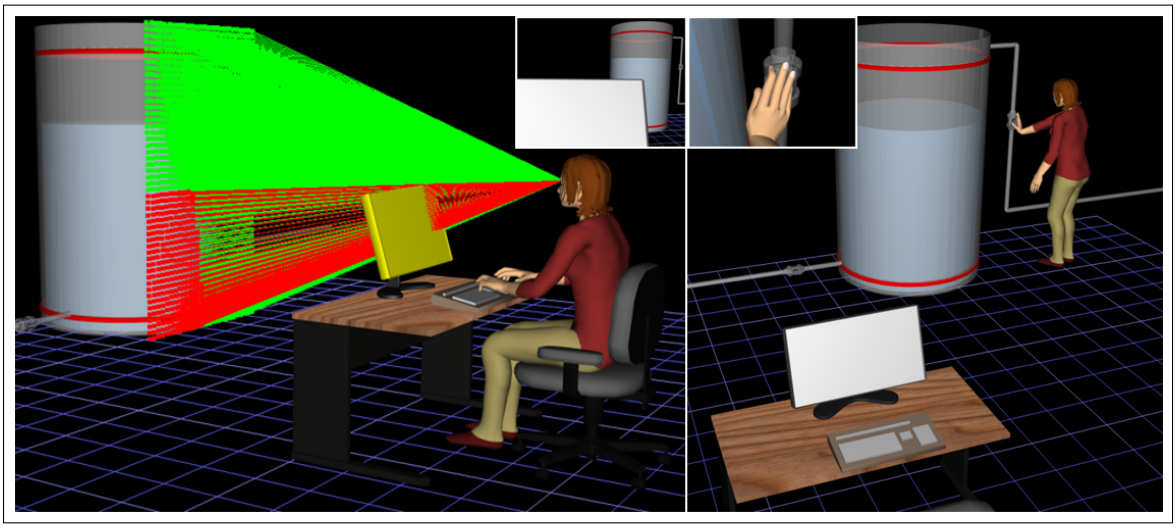


Figure 3.8: DHM vision analysis showing the obscuration zone (left) and reach analysis showing that the valve is accessible (right)

etc.). Figure 3.8 shows a vision and reach analysis for the modified environment with additional work objects/elements. The thumbnails in the figure represent the head forward eye windows of the DHM manikin. The vision analysis shows that the monitor obstructs the vision field of the user and only a part of the tank is visible in this new environment. However, there is enough visibility to detect low and high water levels. Then, using HEFFR, a new set of scenarios can be simulated to assess the potential functional failures, human errors, and the propagation paths of the modified system. A designer may choose to iterate through this process until a satisfactory design is derived.

3.3.4 Discussion

The two scenarios presented above show how the HEFFR framework can be applied to assess functional failures, human error, and their propagation for specific events. In the first scenario, the framework shows how an incorrect valve choice can go on to affect the system, giving the designer insight into the importance of the operator reaching the correct valve. The next scenario shows how a lack of communication between operators and failure to follow procedures could lead to failure.

The results show the capability of HEFFR to assess the propagation paths and the system level effects of potential functional failures (similar to FFIP) and human errors. HEFFR can pinpoint what specific actions the designers need to be considering and what possible human failure they need to try and mitigate.

Moreover, the HEFFR framework does not assume a single user, and the propagation is not based on one human error. The framework allows the modeling of multiple human operators and combinations of possible human errors at the same time. Also, the model allows for scenarios with combinations of function/component failures, cognitive (for instance, failing to detect), and non-cognitive human errors (for instance, failing to grasp). The capabilities of this framework permit the realistic representation of complex event scenarios and the analysis of the interaction between human errors and functional failures and their system-level impact. Overall, the HEFFR framework provides designers/analysts a means to assess both potential functional failures and human errors and how they propagate to affect the system in specific event scenarios at an early design stage. The functional failures, human errors, and the propagation paths are then presented to the designers for further design refinement. Based on the data, a designer may choose components, add functions, or suggest testing to mitigate functional failures. In addition, using HFE principles they may create specifications, suggest operating procedures, or recommend training to mitigate human errors.

The results also shows the advantages of coupling the HEFFR method with DHM analysis tools for non-cognitive human action related analyses. One advantage is that it allows for a visual representation of the human-system interaction at an early design stage. This visual representation allows the designers to visually inspect human-system interactions and come up with human actions to be included in the ASG. Another benefit is that it permits the inclusion of ergonomics analysis of the tasks performed into the HEFFR system model. This ability to perform ergonomic analyses allows the designers to consider ergonomics starting from early design stages. Additional analyses such as reach and vision are also made available to guide designers in their decision making. Overall, coupling DHM with HEFFR analysis acts as a means to apply HFE principles to mitigate potential human errors before any design commitments are made.

There are a few areas where the HEFFR framework can be improved. First, the accuracy of the action simulation is highly dependent on the ASG and the modeling of action simulation step 2. Unlike the action classifications, action outcomes, and action simulation step 1, action simulation step 2 will vary system to system depending on how the system is designed. In the early design stages, defining all possible human interactions exhaustively for each component that requires human intervention and modeling how these actions affect the specific components in action simulation step 2 might be challenging. Second, HEFFR is not capable of capturing ergonomics-related problems when applied alone. However, when coupled with DHM this issue can be addressed. In addition, HEFFR does not capture potential harms that the human could experience. Also, the scenario generation is done manually by the user. This means that the scenarios ran will depend on the user and be highly subjective. It is highly unlikely that any one person or a group can capture all possible scenarios that a complex system could go through in its lifecycle.

In summary, HEFFR provides a systems level modeling approach that allows designers to identify both human-system interactions and system-system interactions that could lead to failures and guide them towards improved system designs at early design stages. In conjunction with HEFFR, DHM can be used as a means to interpret the human-system interaction data to visualize better and analyze the physical aspects of human interactions starting from early design stages until a final design is derived.

3.4 Validating the Failure Prediction Framework

In section 3.3, HEFFR was applied to a simple hold up tank case study. The hold-up tank only consisted five components; a tank, inlet pipe, outlet pipe, inlet valve, and an outlet valve. Even though this simple case study was sufficient to demonstrate the capabilities of HEFFR, its application to a complex design problem is yet to be demonstrated. Also, the accuracy of the failure prediction algorithm needs to be validated. In this section, we address these issues by applying HEFFR to an aircraft design problem and performing a preliminary validation study. First, the subsystems that

played a critical role in the Air France 447 crash are partially modeled. Then, the series of events that led to the accident are fed into the HEFFR framework and the outcomes are assessed to see if they match with what happened in reality. Next, a set of fault scenarios are injected into the framework and into a flight simulator separately, and the results are compared to assess the accuracy of HEFFR. In summary, the HEFFR framework is applied to an aircraft design problem and the results are validated using an accident from the past and a flight simulator software.

3.4.1 Case Study: Air France 447

The Air France 447 crash was chosen as a case study for this research since it was caused primarily due to human error and some component failures. The nature of the crash enables the exploration of the capabilities of HEFFR surrounding human errors, component failures, and their propagation paths. In addition, a comparison between the outcomes of the HEFFR framework and what really happened during the crash can provide some insight into the accuracy of the failure prediction algorithm. In addition to the Air France 447 crash, two additional critical event scenarios are executed within the HEFFR framework and in a commercially available flight simulator separately. A comparison of the results from these executions can further validate the accuracy of the failure prediction algorithm. Overall, the case studies were chosen based on their potential to demonstrate the capabilities of HEFFR surrounding a complex engineered system and the ability act as a validation testbed of the failure prediction algorithm.

The following events are a summary of what contributed towards the Air France 447 crash[131, 132]. The aircraft, an Airbus A330 entered a tropical storm region. The crew chose to fly through the worst of the storm while the surrounding aircraft chose to fly around it. Since the aircraft was flying through clouds, the pilots turned on the anti-icing system to keep ice off the flight surfaces. At this point, the captain had left to take a nap, leaving the copilots at the cockpit. Also, he had left the less experienced of the two in-charge of the controls. The formation of ice crystals in the pitot tubes caused the loss of airspeed measurements. This caused the autopilot system to shut off

and switched the airplane from “normal law” to “alternate law.” It is impossible to stall an Airbus A330 when it is in “normal law” because of how its fly by wire system works[131]. However, when the airplane is in “alternate law,” the aircraft can go into a stall[131]. Failing to understand the situation, the confused copilot in-charge pulled back on the flight control to put the aircraft in a steep climb. This caused the stall warning to go off. However, the copilots failed to acknowledge it and continued to climb causing the aircraft to lose airspeed. Eventually, the copilot who was not in charge of the controls realized the decreasing airspeed and asked the copilot in-charge to pay attention to the speed. At this point, the pitot tubes had started to work again and they started to get valid airspeed information. The copilot in-charge eased on the flight control causing the aircraft to climb at a slower rate and gain airspeed. The stall warning stopped sounding and the copilots gained control of the aircraft.

Then, suddenly for reasons not known, the copilot in-charge pulled back on the stick again, causing a climb and reduction in airspeed. This activated the stall warning again. By this time, all pitot tubes were functional and the avionics of the aircraft were back to normal. The copilot in-charge increased the throttle to TOGA level (take off and go around). This level is used at low altitudes to effectively increase speed and gain altitude during takeoff and go-arounds[131]. Since this was performed at a much higher altitude (37,500 feet), where the air is much thinner, the aircraft did not climb and gain speed as expected. Instead, it started falling towards the ocean. The copilot who was not in charge of the controls had no idea that the other pilot had the flight control pulled back. He was expecting the aircraft nose to be down. This is because the Airbus flight control sticks on either side of the cockpit work independently and give no feedback on how the control on the opposite side is operated. At last, the more experienced copilot took control of the aircraft. However, he did not grasp the stall situation and started pulling back on the control causing the nose to be pitched up. The aircraft continued to fall towards the ocean.

By this time, the captain arrived at the cockpit but made no attempt to take control of the aircraft. Instead, he was observing and giving instructions. The more experienced copilot pushed the stick forward at last. But this was not sufficient to pitch the nose down since the other copilot was

still pulling the stick back and the mode the plane was set in, took only the average of these inputs. Unfortunately, the aircraft continued to fall and crashed. We have identified the side stick (flight control stick), throttle system, auto-pilot system, airspeed indicators, and the stall warning system as the most critical systems that contributed to the Air France 447 crash. In this paper, we have modeled these systems using the HEFFR framework and applied the event sequence described above as inputs and compared the outputs with the outcomes of the crash to first explore the applicability of HEFFR to a complex design problem and to validate the accuracy of the failure prediction algorithm.

In addition to evaluating HEFFR using a real-world example, this research uses a commercial flight simulator to do additional validation studies of the HEFFR framework. X-plane was chosen since it can provide Federal Aviation Administration (FAA) certified simulation and vehicle models[133]. In addition, X-plane allows the simulation of various fault scenarios such as component failures, adverse weather, and cockpit environmental factors that can be activated at any time during the flight. X-plane is regularly updated to keep flight models and world models up-to-date. Also, it has been widely used by researchers for various purposes such as unmanned aerial vehicle (UAV) simulations[133], testing of hardware in the loop miniature-autopilot evaluation system[134], software-in-the-loop simulations for UAVs[135], and as a rendering platform for the testing of heads-up displays in small aircraft[136]. In this research, an Airbus A330 was chosen to perform the simulations so that it is consistent with the other case study. The Airbus A330 used for this simulation is not certified. However, it was still chosen for this simulation because of its affordable price and very realistic flight models.

Two additional scenarios were executed in X-plane and HEFFR separately as supplementary validation of the failure prediction algorithm. In the first scenario, the events that took place in the Air France 447 crash occur. However, instead of pulling the stick back until the flight stalls and crashes, the pilot recognizes his mistake when the stall warning goes off and pitches the nose down. The airspeed indicators continue to malfunction. However, the pilot remembers to check other instruments to determine airspeed. In the second scenario, the flight is cruising with auto-pilot on. The pilot inadvertently hits and moves the side control stick. This causes the auto-pilot to shut off

and the warning to activate. The pilot fails to recognize that the auto-pilot is shut-off and ignores the warning. The sudden drop in altitude and increase in speed confuses the pilot, but they think that it's coming from unreliable sensor data. Next, they contact air traffic control for altitude and airspeed data, and once they receive the feedback, with closer inspection, they realize their mistake and correct it. These scenarios were chosen because they involve the same subsystems as the ones that were modeled for the Air France 447 crash and they present enough complexity to challenge the failure prediction algorithm. In the following section, details on how to setup the problem and the simulation are provided.

3.4.2 Methodology

The purpose of this research is to apply the HEFFR framework to an aircraft design problem to demonstrate its application in complex engineered systems design and to evaluate the accuracy of the failure prediction algorithm. The accuracy of the failure prediction algorithm is validated using two approaches. First, the events that led to the Air France 447 crash are fed into the HEFFR framework and the outcomes are compared with the events from the actual crash. Second, critical event scenarios surrounding pilot error are formulated and fed into the HEFFR framework and a commercially available flight simulator separately, and the outcomes from both are compared. This section goes into detail on how to setup the problem using the HEFFR framework. HEFFR uses the exact same modules (Functional Model, Configuration Flow Graph (CFG), Behavior Model, Behavior Simulation, and FFL from FFIP to track functional failures and their propagation paths. Note that this research does not go into detail on how to create modules from FFIP since these are fairly well documented in the past [23, 24, 92, 119, 120].

3.4.2.1 System Representation

The first step in HEFFR is to create a system model. The system model includes a functional model, CFG, Behavior Model, and ASGs. The subsystems that played the most critical role in the Air France 447 crash are the side stick, the throttle system, the autopilot system, airspeed indicators, and the stall warning system. As shown in Fig. 3.9, the functional model and the CFG only partially represent these subsystems because creating full-scale models will not add value to this research. Functional models and CFGs are used in the behavior simulation and FFL during the evaluation of functional failures and their propagation paths. As mentioned earlier in this chapter, these segments (functional models, CFG, behavior simulation, and FFL) in HEFFR come directly from FFIP and have been fairly well tested and validated[23, 24, 92, 119, 120]. Hence, spending time on elaborating and validating these modules will not add value. Instead, this research focuses more on the human error related aspects of the HEFFR framework so that the new modules are tested and validated.

The parallel lines crossing over the arrows in Fig.3.9 indicate that there is a discontinuity in the graph. For example, the flight data arc that goes into the flight computer node in the CFG has a discontinuity because the data gathering methods are not represented in the graph. The next step in the framework is to identify the components that interact with the humans and create ASGs for them. For this study, the flight computer, throttle lever, and the control stick were identified as the components that require ASGs. Even though displays and warnings are components that interact with humans, they are not identified as components that require ASGs because these components act as control signals for the components control stick, throttle lever, and flight computer. Hence, the interaction between these components and the human are already represented in the ASGs. In addition, the ASG for the flight computer is not modeled in this study because the Air France 447 crash did not have any critical events in which the flight computer and the pilot had direct interactions. Figure 3.10 shows the ASGs for the control stick and the throttle lever.

Next, each action in the ASGs need to be classified to represent all nominal and faulty states. For example, the action Grasp Control Stick can have the following classifications.

- Nominal: Grasped the reached object.

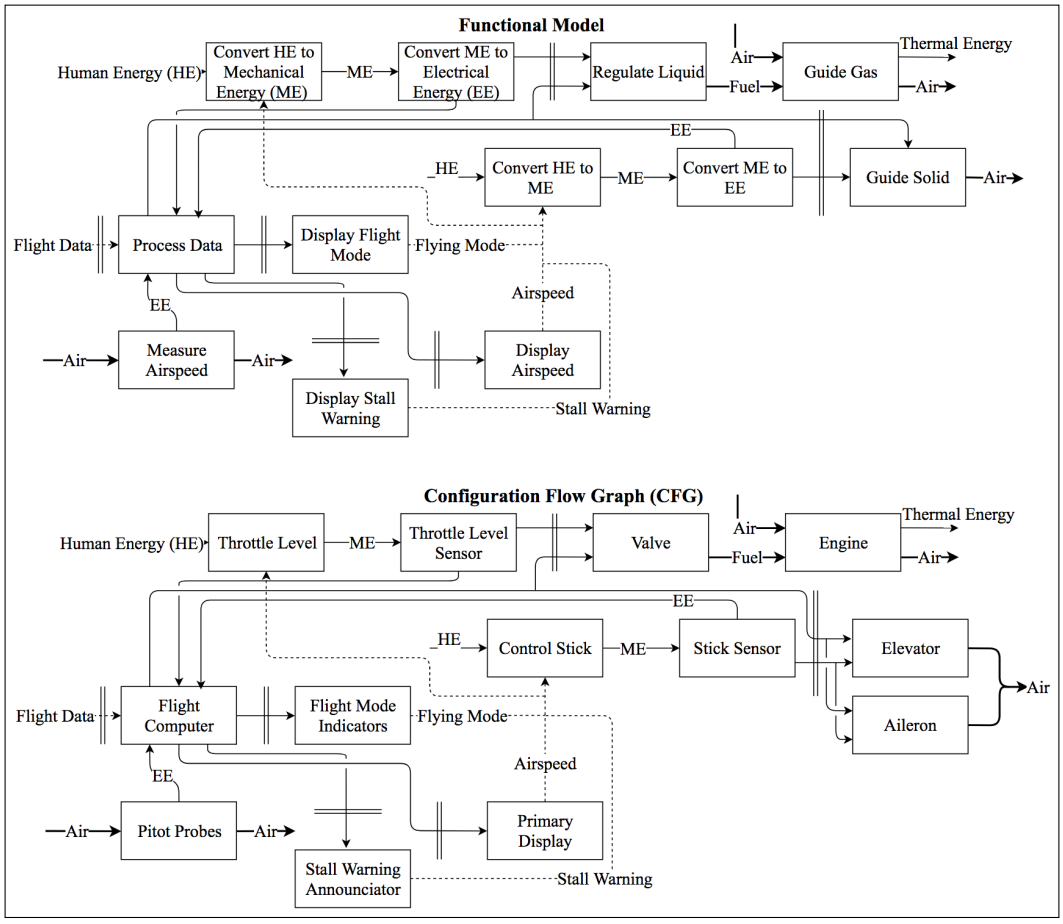


Figure 3.9: Partial Functional Model and Configuration Flow Graph of the subsystems that played a critical role in the Air France 447 crash

- Failed: Unable to grasp the reached object.
- No Action: The action was not attempted.

The action classifications for the same action regardless of what component they are applied to can be re-used across the system. Table 3.4 shows the classifications for the actions from the ASGs. To create the behavior model the human induced and non-human induced behaviors should be defined for components that interact with human. The Control Stick has “nominally activated,” “nominally inactive,” “falsely activated,” and “falsely inactive” as human induced behavior modes

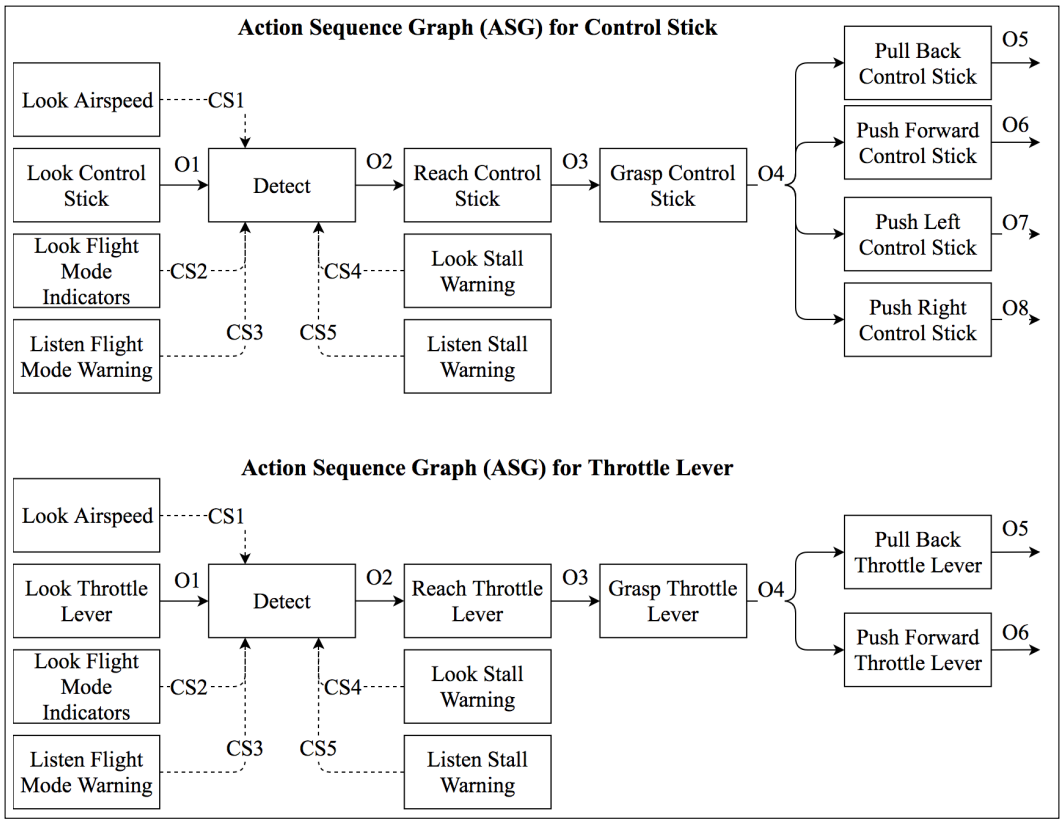


Figure 3.10: Action Sequence Graphs for the control stick and the throttle lever

where active means either Pushed Forward, Pulled Back, Pushed Left, or Pushed Right. The non-human induced behavior modes are "stuck active" and "stuck inactive." The throttle lever has the exact same behavior modes as the control stick. However, active in this case means the following four levels; Idle Level, Climb Level, Flex and Mac Continuous (FLX MCT) Level, and Takeoff and Go Around (TOGA) Level. Similarly, the behavior modes for the components that do not interact with human are then defined. With the Functional Model, CFG, ASG, and Behavior Models completed, the system model is now fully developed.

Table 3.4: Action classifications for the actions represented in the Action Sequence Graphs

Actions	**Nominal and Faulty Responses**			
Look	Saw	Cannot See		
Hear	Heard	Cannot Hear		
Detect	Detected CSX- Nominal	Not Detected CSX -Nominal	Detected CSX – Failed	Not Detected CSX - Failed
Reach	Reached – Nominal	Reached - Failed	Cannot Reach	No Action
Grasp	Grasped	Cannot Grasp	No Action	
Pull	Pulled	Cannot Pull	No Action	
Push	Pushed	Cannot Push	No Action	

3.4.2.2 The Simulation

To begin the simulation, the action simulation has to be set up. In the first step, the action classification for each action is determined based on the critical event scenarios using the input-output relationships for each action. For example, for the action Grasp Control Stick, if the input (O3) is equal to "reached" and the output is equal to "grasped," the action is classified as "grasped - nominal." If 03 equals "not reached" then it is classified as "no action" since its impossible to grasp an object that is not reached. Similarly, the input-output relationships for each action need to be defined so that the action simulation can determine the classifications for each action in the ASG. In the next step, the action simulation tracks the evolution of action classifications using the ASG to determine the human-induced behavior modes of the components. For example, for the control stick, if the flight mode is "autopilot off - alternate law," the stall warning is "on," Reach is classified as "reached - false," and all Pull and Push actions are classified as "no action," the behavior mode will be set to "falsely inactive." This is because the pilot is expected to pitch the nose down when the stall warning is "on" and that does not happen in the scenario presented. Similarly, all human-induced behavior modes are defined for all components so that this step of the action simulation can accurately determine the behavior modes based on the critical event scenario inputs.

Note that the Airbus A330 can receive independent inputs from the control sticks from either side of the cockpit. Two control stick inputs are recorded and simulated on each time step to simulate the inputs from both pilots. However, the number of control stick inputs are reduced to one based on the control stick mode configuration during the behavior mode determination. For example, if the flight mode is set to take only the average of both control stick inputs, one input is "pushed forward," and

the other one is “pulled backward,” the simulation will calculate the stick to be inactive. Similarly, if the flight mode is set to take inputs only from one of the two control stick, the inputs from the control stick that is not set as input will be ignored during the behavior mode determination step of the action simulation.

The behavior simulation determines the non-human induced behaviors of the components by tracking the input-output relations of each node in the CFG. For example, the flight control stick will be determined as “stuck inactive” if there was some mechanical failure in the stick that was causing it to not move. The FFL takes the behavior modes from both action simulation and behavior simulation to identify the health of each function in the functional model. The functions are determined as “lost,” “degraded,” or “operating.” Overall, the framework takes critical event scenarios as inputs; uses the ASG, action classifications, and action simulation to determine human induced behavior modes of the components; and uses the CFG, behavior model, and behavior simulation to determine the non-human induced behaviors of the components. The action simulation and the behavior simulation happen in parallel at each time step and the resulting behavior modes are fed into FFL to determine the functional health of each function and overall system. In the following section, we discuss the results of the executions of the 3 scenarios from section 3.4.1.

3.4.3 Results

First, the events that contributed to the Air France 447 crash was executed. To begin the simulation, all functions, components, and actions were initialized to reflect the cruising phase the aircraft was in before the crash. For example, flight mode, air speed, and stall warning were set to “autopilot-normal law,” “nominal,” and “off,” respectively, and most of the human actions were set to “no action.” The actions were set to “no action” because at this point the autopilot was on and the pilots did not have to interact with any of the components. However, the action Detect was set to whatever the detection state the pilot was in for each control signal. For example, for the control signal of the flight mode, the action Detect was set to “detected - nominal” since the pilots recognized that they were

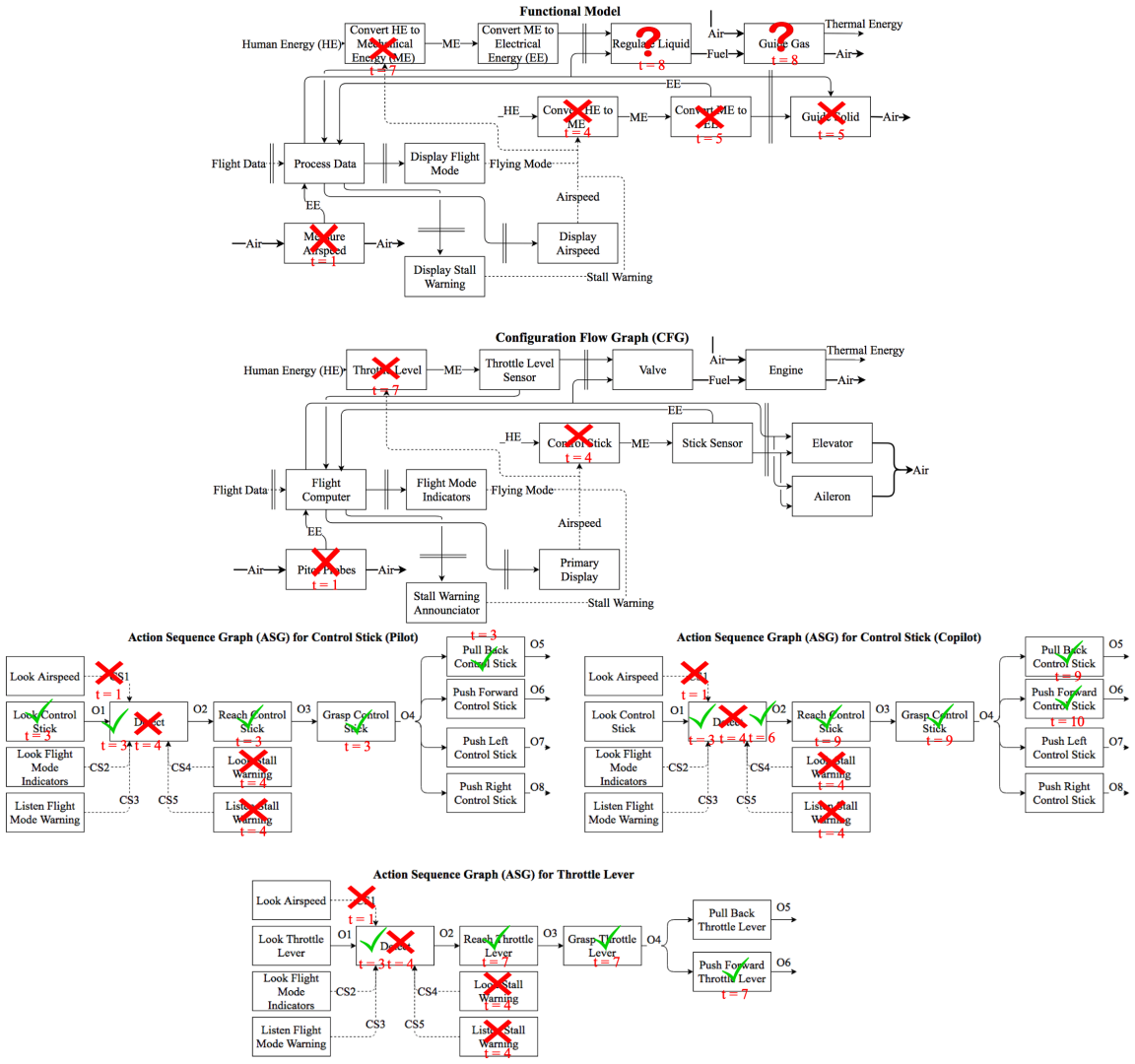


Figure 3.11: Results of Human Error and Functional Failure Reasoning framework for the execution of scenario 1 (Air France 447)

in autopilot and did not attempt to control the aircraft. The events that occurred before the loss of air speed measurement cannot be modeled in HEFFR because they were either weather related events or involved movement in the cockpit. Hence, at time step (t) = 1 behavior of the pitot tube is set to “failed” to reflect the frozen pitot tube. This causes the function measure airspeed to be lost.

At t = 2 the flight mode is set to "autopilot off - alternate law." At t = 3, the pilot confusion is recorded by setting the actions Detect, Reach Control Stick, Grasp Control Stick, and Pull Back Control Stick as successful executions for the pilot because at this point the pilot recognized that there was an issue with the airspeed and did not have any trouble accessing the stick to perform his intended action. At t = 4, the stall warning is set to "on" and the action detect to "not detected - failed CS stall warning." This caused the action simulation to determine that the control stick is in "falsely activated - pulled back" behavior mode. This results in function Convert HE to ME to be lost. In the next time step, functions Convert ME to EE and Guide Solid are lost. At t = 6, the copilot's recognition of loss of altitude is represented by setting the action Detect for the copilot to nominal. However, the pilot easing on the control stick cannot be represented since HEFFR only takes discrete values and in this case it will be either pulled back or not. So the system continues to stay the same. This is different in reality because the pilots were able to briefly gain control of the aircraft at this point. The pitot probe starts to function again in this time step. This results in the function Measure Airspeed being restored.

Next, the throttle is moved to TOGA level. This is represented by setting the actions Reach, Grasp, and Push Forward to successful execution because the pilot had no issues with performing the intended functions. However, the action simulation determined that the throttle is in "falsely activated - TOGA level" behavior mode. At t = 8, functions Regulate Liquid and Guide Gas become degraded because of the throttle being set to TOGA level. Then, the co-pilot takes control of the aircraft. But he too pulls back on the flight stick. This causes no changes to the system. At t = 10, the action Push Forward Control Stick for the copilot is set to "nominal." Since the flight mode the aircraft was in at this point takes the average of the two controls, the behavior of the flight stick is determined to be "falsely inactive." The results of this execution is shown in Fig. 3.11 where ✓means nominal execution, ✗means non-nominal operation or loss of function, and ? means degradation of function. The time steps in which these events occurred are also listed below each sign.

The following two scenarios were executed in HEFFR and X-plane separately. For the first scenario, the HEFFR execution is the same as the Air France 447 scenario until t=5. However,

at t=6, the pilot's recognition of the mistake and his ability to determine the airspeed using other instruments in flight is represented by setting the action Detect to "nominal." In the same time step, the pilot pushes the control stick forward to gain speed and recover from the stall. These results in control sticks behavior to change to "nominally active." The functions Convert HE to ME, Convert ME to EE, and Guide Solid are restored in the following time step. The stall warning is shutoff in this time step. The function Measure Airspeed continues to be lost. However, this does not propagate to affect the system overall because of the pilots detection of this and the corrective measure taken. The X-plane execution of this scenario started with the take off and subsequently entering the cruising phase. Then, the pitot tube failure is induced. The initial actions of the pilot cause some flight instability. However, when the pilot realized the mistake and took corrective measure, the pilot was able to gain full control of the aircraft. Figure 3.12 shows the primary flight display during the instability, during the corrective actions, and after the flight was stable. The pitch angle and the vertical speed (highlighted using white dashed lines in the figure) indicate that the nose was up (positive pitch angle) and still the flight was sinking (negative vertical speed) during the instability; nose was down and still the flight was losing altitude during the corrective measure; and nose was slightly tilted up and the flight held a steady altitude (near zero vertical speed) when stability was gained.

For the next scenario, the simulation is initialized similar to the Air France 447 scenario at t =0. At t=1, the inadvertent activation of the control stick turns off the autopilot. Since the pilot did not recognize it, the action Detect is set to "failed." In the next time step, the the behavior mode of the control stick becomes "falsely inactive" because no action was reported in the ASG to indicate that the pilot was taking control of the stick. In this time step, the pilot's confusion with the air speed is represented by setting the detection related to air speed to "not detected CS airspeed - failed." When the air traffic data was received, the pilots realization of the mistake is represented by setting the action Detect to "nominal" at t =3 and actions Reach, Grasp, and Push/Pull for the control stick is set to completed (i.e., "reached - nominal," "grasped," etc.). This causes the behavior mode of the control stick to go to "nominally active." In this case no functional losses were recorded and the

Figure 3.12: Primary Flight Display (PFD) during the stall, stabilization, and stable stages of X-plane flight simulation for the first scenario

system continued to act nominally after this time step. In the X-plane simulation, the events are initialized during the cruise phase of the flight. When the stick is moved, the autopilot shuts off and the warning sounds. The aircraft starts to loose altitude with the nose slight pointing down. This causes an increase in airspeed. When the air traffic control provides airspeed and altitude data, the flight is brought back to the cruise altitude and autopilot is turned on.

3.4.4 Discussion

The scenarios presented above show how HEFFR framework can be applied to a complex engineered system to identify potential human errors, component failures, and their propagation paths. In the first scenario, the temporary loss of airspeed indicators is represented by the temporary loss of the function Measure Airspeed. The human error of failed detection (or confusion) propagates into several functional losses for the overall system. In the next scenario, the same failure and errors occur but the recognition of the events (situational awareness) and the proceeding actions helped the system recover from the failures and human errors experienced earlier. In the final scenario, the inadvertent activation of a control causes some confusion and results in a temporary none nominal state. The potential for failure due to accidental activation of a control is demonstrated and the importance of designing the control in a way that is foolproof from being activated accidentally is

stressed. Having this amount of insight early during the conceptualization stages of design will help designers come up with better designs that are less likely to fail due to component failures and/or human errors. In addition, the designers may suggest testing, training, and/or operation procedures to mitigate the potential risks.

During the execution of the Air France 447 crash, the action in which the pilot eased on how much he was pulling on the control stick was not represented in the HEFFR framework. This was because the framework only takes discrete inputs and for the control stick, the inputs were restricted to either being pulled or not. However, this restriction didn't limit the ability of the framework to represent the overall system level failures. In addition, the events that contributed to the crash had to be simplified in order to be represented in HEFFR. Similarly, the results from HEFFR was simply indicating the functions that were lost or degraded. It did not directly point to a stall. But one could infer that when the function Guide Solid was lost, the pilots lost control of the aircraft indicating a potential stall. Additionally, the loss of airspeed data and the return of it was reflected in HEFFR. The function Measure Airspeed was lost and then recovered as the events progressed. Being able to accurately represent temporary failures is critical. This allows designers to assess if any mitigating actions or recovery procedures are needed for the temporary failures. The control stick did not have any failures associated with it. However, it was in a faulty state due to the human actions for the majority of the execution. This ability of HEFFR, allows designers to understand how human actions can affect components regardless of how reliable they are.

Similar to the Air France 447 scenario, the scenarios executed in the flight simulator had to be simplified in order to be analyzed using HEFFR. While the outcomes of HEFFR lacked details when compared to the outcomes of the flight simulator, they gave enough details for a designer to consider re-design or mitigating actions. For example, in the first flight simulator scenario, the pilot correcting the stall and operating the flight without airspeed measurements was represented in the HEFFR framework with function Guide Solid becoming operational and the stall warning shutting off. Also, the pilot's actions turned nominal as detection of airspeed became nominal. Likewise, in the second flight simulator scenario, even though no adverse consequences were seen in the flight simulator,

HEFFR framework was able to identify the temporary mishap of the pilot and the propagating consequences. The results from both of these executions can give insight into the potential corrective actions that need to be taken to mitigate the risk when a failure or human error occurs. With such information, designers can suggest training or operating procedures so that users are not caught off guard when such errors/failures occur.

Overall, one could argue that HEFFR lacks fidelity when compared with events that happen in reality or high fidelity simulations. However, this is an expected trait in a framework like HEFFR, because it is intended to be applied at the conceptual design stage where only a limited amount of information is available. Usually, detailed information on how exactly a system is modeled and how the intended functions will be fulfilled only become available as the design evolves into later stages design. Predicting component failures, human errors, and their propagation paths as it would happen in reality becomes highly impossible if relevant data is not available. However, the ability to identify potential component failures, human errors and their propagation paths with enough details to give insight to the designers to mitigate the potential risk and build reliable systems is more important than producing high fidelity models. All three scenarios presented in this research show the capability of the HEFFR framework to achieve this. In addition, they also show that HEFFR is capable of identifying faulty behavior states of components even when there is no failure present. Imagine that the engineers at Airbus had a tool similar to HEFFR. Given that they would have had an opportunity to execute the three scenarios presented in this paper, there is a probability that Air France 447 would never have happened. They would have identified the potential for the Air France 447 crash through the first scenario. The second and third scenarios would have given them insight into the importance of training the pilots to face such events.

3.5 Conclusion

This chapter introduced the Human Error and Functional Failure Reasoning framework to analyze potential functional failures and human errors in complex engineered systems during early design

stages. The ultimate goal of this research is to combine both functional failure analysis and human error assessment and determine failure propagation paths during early system design. This way, the design teams can comprehensively explore potential risks and failures related to both components/-functions and human errors before any design commitments are made to design systems that are effectively guarded against such risks and failures. The HEFFR framework was applied to a hold-up tank problem to demonstrate how it can be applied towards designing more reliable systems. Digital human modeling can be coupled with HEFFR as a tool to interpret non-cognitive musculoskeletal and vision related ergonomics of the human-product interactions and to explore design alternatives and ergonomic issues. Additionally, this research showed the application of the HEFFR framework on a complex engineered system and performed a validation study of the failure prediction algorithm.

As part of the validation study, we applied the HEFFR framework to an aircraft design problem by modeling the most critical subsystems that played a role in the Air France 447 flight crash. We then compared the results from HEFFR with what happened in reality to validate the failure prediction algorithm. We executed two additional failure scenarios in a commercially available flight simulator and HEFFR separately and compared the results between the two as additional validation of the failure prediction algorithm. The results from the comparison studies show that the HEFFR framework is capable of predicting potential failures realistically but lacks fidelity. Future work can look into how this method can be adapted to be used in later design stages so that designers can use it throughout the design process until a final design is approved. This can be achieved by introducing continuous variables for the behavior modes and action outcomes. For example, the action Pull Backward may take the distance between the new position and original position as an outcome so that the exact pitch angle can be determined. This will allow a high fidelity computational failure prediction simulation.

Chapter 4: Identifying Worst-case Fault Scenarios

This Chapter addresses research objective 2 by expanding the Human Error and Functional Failure Reasoning (HEFFR) framework introduced in Chapter 3 to enable designers assess a majority of potential fault scenarios by formulating a scenario generation and fault quantification model. The automated scenario generation approach utilizes techniques used in automated test case generation in software engineering to generate a wide range of potential fault scenarios involving humans and components. This research was published in the ASME Journal of Computers and Information Science in Engineering and in the Proceedings of the 2019 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference and was coauthored by Lukman Irshad, H. Onan Demirel, and Irem Y. Tumer [137, 138]. Next, risk quantification model based on expected cost, component failure rates, and human error probability is introduced to help designers quantify the severity of failures and prioritize worst-case fault scenarios. This research was published in the Journal of Mechanical Design and in the Proceedings of the 2020 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference and was coauthored by Lukman Irshad, Daniel Hulse H. Onan Demirel, Irem Y. Tumer, and David C. Jensen [139, 140].

4.1 Motivation

The HEFFR framework introduced in Chapter 3 requires fault scenario inputs, which motivates the research discussed in this chapter. To be able to mitigate potential risks effectively, one needs to assess a majority of potential fault scenarios to identify the worst-case fault scenarios and take mitigating action. Hence, to perform a comprehensive analysis, the HEFFR framework requires input scenarios that cover a majority of if not all component failures and human errors. A major

shortcoming of the HEFFR framework is that it relies on the designer(s) to come up with such use cases; thus, making it highly subjective. Also, it is highly unlikely that anyone or a group can capture use cases exhaustively enough to cover a majority of the fault scenarios. While assessing a broad spectrum of fault scenarios can give some insight into potential risk, without quantifying the failures, there is no way to prioritize fault scenarios. Without prioritization, designers will have to treat all scenarios equally when considering mitigating action, which is unrealistic when there is a large number of potential fault scenarios.

This chapter addresses the above gaps by introducing an approach to automatically generate fault scenarios and deriving a model to quantify the risk of failures resulting from interactions between human error and component failures. The goal of the automated scenario generation approach is to generate use cases that can cover a wide range of fault conditions involving both component failures and human errors. The risk quantification model aims to capture the severity of failures that are caused by human and component fallibilities acting in tandem. The overall goal of this research is to allow designers to assess the risk of hazards emerging from human- and component-related failures occurring in combination and identify worst-case fault scenarios.

The automated scenario generation technique uses unified modeling language based automated test case generation techniques from software engineering as a basis to automatically generate a wide range of potential fault scenarios involving humans and components to be used as inputs for HEFFR. It uses a modified Depth First Search (DFS) algorithm, component behavior models, and the human action classifications to achieve its intended purpose. This approach creates scenarios that cover all potential behaviors of both components and humans to make sure that a majority of potential fault scenarios are produced. In reality, failures are not limited to one failure event. At times, when a failure event occurs, it may go unnoticed. Such failures may come to light when other failures are detected. Hence, the scenarios generated through this approach include multiple failure events to represent what may happen in reality, reasonably. Furthermore, when a failure occurs, they can cause other failures. To represent the cascading effects of failures, the fault scenarios allow failures to propagate for a user-defined number of time steps. Finally, when component failures occur, they

do not return to a nominal state unless they are repaired. On the other hand, when human errors occur, humans can correct their errors and return the system to a nominal state. The automated scenario generation technique takes this phenomenon into consideration to portray what may happen in reality. In summary, the automated scenario generation technique aims to generate a wide range of potential fault scenarios involving human and components in a way that they are representative of what may happen in reality.

The risk quantification approach uses a cost and probability model to quantify the relative impact (and thus priority) of critical event scenarios. To calculate the likelihood of the occurrence of critical events, this research considers both component failure and human error probabilities, using traditional reliability engineering principles to estimate component failure probabilities and the Human Error Assessment and Reduction Technique (HEART) [25] to estimate human error probabilities. To quantify the relative importance and priority of failures, this research adapts the expected cost of resilience metric developed in Ref. [26], which defines expected cost as the multiplication of the modeled probability and cost of the scenario. Designers can use these metrics with the automated scenario generation technique to identify worst-case fault scenarios, prioritize fault scenarios, quantify the impact of human errors and component failure, and pinpoint areas (both component and human interaction related) where improvements can yield the greatest risk mitigation. Additionally, the risk metrics will allows the use of the HEFFR framework to perform risk-based trade-off studies, identify points of automation, select appropriate components, and establish operating procedures, training, and safety protocols.. However, models of risk in these systems have implications to how best to account for risks in the design process, and may additionally be subject to model and parameter uncertainties. Thus, designers need to account for these uncertainties when modeling human-component faults using this model.

4.2 Background

This section explores automated scenario generation methods in engineered systems design and software engineering to form the ground work for the automated scenario generation method introduced in this research. Then, the methods used to quantify the probability of failure and human error are discussed to build the foundation for the likelihood of occurrence calculation. Finally, the theoretical basis for using expected cost in risk quantification is explained.

4.2.1 Automated Scenario Generation for Complex Engineered Systems Design and Failure Assessment

Previous research has attempted to automate scenario generation to conduct failure assessment and to validate system designs. One such attempt automatically generates test cases to validate system design and implementation against requirements using a four-part algorithm [141]. First, the algorithm uses the requirements model and the Greedy Search algorithm to identify base scenarios that have the potential to test all requirements. Next, incomplete base scenarios are identified and enhanced to make them complete. Finally, the base scenarios and the enhanced base scenarios are combined to create a comprehensive list of test cases. TestWeaver [142] is another automated test case generation tool used for systematic testing. It works like a game of chess where TestWeaver plays against the system under test by making a series of moves with the goal of attaining goal states which force the system to violate requirements. This allows the testing of a wide range of alternative paths that can contribute to requirement violations. This tool was successfully used in the design of the crosswind stabilization function to the Active Body Control (ABC) suspension for the Mercedez-Benz 2009 S-Class [143]. Both of the test case generation methods discussed above do not explicitly search for failures instead they are intended towards identifying requirement violations.

There have also been attempts to automate failure assessment by automating the fault cause and effect scenarios. Prior research has implemented automated failure cause generation in Failure Modes

and Effects Analysis (FMEA) for diagnostics and prognosis analysis. One such method induces various component failures to the system and compares results between nominal system behavior and faulty system behavior to understand the effect of a failed component [144]. The algorithm performs this analysis by exhaustively covering all possible component failures. Another diagnostic application of FMEA automatically generates diagnostics and fault analysis [145]. Another study looks into improving reliability by automatically generating fault trees. It uses a Finite State Machine (FSM) based system model to generate a fault tree that consists of all possible failures[146]. Another method aimed at mitigating risk, SimpraPlan[147], uses functional requirements and the physical structure of the system to generate scenarios that test for system vulnerabilities. There are several other methods that automate event tree generation. However, these event tree generation methods either do not directly relate to system design [148, 149] or require historical data or detailed system data [150–152], making them inapplicable during early design stages. All of the methods presented above either do not apply in a design context, do not generate human-related fault conditions, or require detailed system data. Hence, they are not effective when it comes to generating use cases for HEFFR analysis.

Inherent Behavior of Functional Models (IBFM) [153] uses functional models and functional behavior models to automatically generate fault scenarios to assess potential failures and their system-level effects. The simulation starts with introducing one fault at a time and progresses by incrementing the number of faults introduced by one. It also allows for pseudo-time based simulations. Even though IBFM can be applied early in design, since it does not consider component behavior models, the scenario generation technique cannot be applied to HEFFR. Another early design stage failure assessment framework, Functional Failure Identification and Propagation (FFIP), has an extension in which automated scenario generation is present [154]. It generates event trees for triggering events that fail to activate a set of predefined safety functions. This method can only evaluate one triggering event and the corresponding event tree at a time; thus, making the overall analysis human resource intensive when a large number of triggering events need to be analyzed. Both of these methods are not capable of generating human-machine interaction related use cases

Table 4.1: Comparison between risk assessment methods with automated scenario generation and the proposed automated scenario generation method in this research

	Ref. [144]	**Ref. [145]**	**Ref. [146]**	**SimpraPlan[147]**	**IBFM[153]**	**[154]**	**HEFFR**
Ability to generate scenarios relating to components	✓	✓	✓	✓	✓	✓	✓
Ability to generate scenarios relating to human	✗	✗	✗	✗	✗		✓
Applicability for Risk Assessment	✓	✗	✓	✓	✓	✓	✓
Applicability during early design stages	✗	✗	✗	✗	✓	✓	✓
Ability to generate a majority of fault scenarios involving both human and components	✗	✗	✗	✗	✗	✗	✓

with enough detail to be applied in a HEFFR analysis. As none of the techniques detailed above provide a sufficient solution to generate use cases for HEFFR automatically, this research explores other fields that utilized model-based system representations. A comparison between the proposed work and the existing risk assessment methods with automated fault scenario generation is provided in Table 4.1.

4.2.2 Automated Scenario Generation in Software Engineering

We have explored the Model Based Testing (MBT) methods used in software engineering because they involve automated test case generations. The advantages of MBT are listed below [155].

- It can reduce design cost.
- It provides the ability to identify issues with requirements.
- It allows testing early in the software design lifecycle.
- It allows for comprehensive tests that exhaustively cover all potential use cases.
- Fault Detection is more effective and efficient when compared with other types of software testing.

The advantages of MBT are characteristics that would be ideal in the use case generation method of HEFFR. Hence, we explored the test case generation methods utilized in MBT further. One such test case generation method uses environmental behavior for scenario generation [156]. It defines

the behaviors of the system using event traces that are made of relations between precedence and inclusion. Event grammars, which specify the possible event traces, are traversed top-down and left-right to generate test cases and evaluate cyber-physical systems. Another test case generation framework uses high-level Petri Nets [157] to generate functional models, access control models, and potential threat models. Petri Nets is a systematic method to model and verify software systems [157]. The Petri Net models are then searched using Depth First Search (DFS) and Breadth First Search (BFS) to generate test cases automatically. A type of Petri Net model (namely Coloured Petri Net model) that can an be used to model distributed systems is used in another approach to generate test cases for distributed system protocols [158]. This approach takes a simulation-based approach to automatically generating test cases. In an attempt to overcome challenges relating to testing and verifying dynamic Simulink models, Matinnejad et al. [159] proposed a meta-heuristic search based test case generation method that covers both continuous and discrete behaviors. The test case generation aims to increase the diversity in the output signals so that the chances of finding unexpected output signals are maximized. Finally, the generated test cases are prioritized based on their likelihood of identifying faults.

Unified Modified Language (UML) based automated test generation methods are commonly used MBT types [160]. UML is a system modeling language that includes several diagrams to represent the architectural and behavioral aspects of a system [161]. Because of its wide use, usability, and effectiveness, test case generation methods based on UML are highly popular [160]. One such method [162] uses state charts to create FSMs using a tool called PerformCharts. Then, the FSMs are fed into Condado, a graph theory based test case generation tool, to automatically generate test cases that cover all possible transitions. A similar approach converts state charts into an intermediate graph which is then traversed based on various coverage criteria to come up with test cases [163]. Another method [164] extracts data from class diagrams, sequence diagrams, and state diagram to automatically generate test cases. Swain, Mohapatra, and Mall [165] proposed a framework that combines state models and activity models to create a state activity graph which is then searched using DFS to generate test cases.

Numerous UML based test case generation methods use activity diagrams as the basis to generate test cases. For instance, one method uses an exhaustive search and a test queue prioritization technique to identify critical test cases [166]. Another method combines Tabu Search with test cases originating from activity diagrams to generate test cases [167]. Stallbaum, Metzger, and Pohl [168] used risk-based prioritization to generate test cases. The EasyTest method coverts activity diagrams to activity dependency tables, and then into activity dependency graphs, and traverses using a dept-first search based algorithm to come up with test paths and subsequently test cases[169]. Another approach to UML based test case generation is to utilize use case diagrams as the basis to generate the test cases. For example, one technique utilizes use case simulations to build test objectives and sequence diagrams to generate test cases from test objectives [170]. Another technique uses use case and sequence diagrams to create a system testing graph which is then traversed to generate test cases [171]. Raza, Nadeem, and Iqbal [172] proposed a framework that uses the Interactive Overview Diagram and a series of matrix generations to generate test cases for a specific coverage criteria. Prasanna and Chandran [173] came up with a framework that uses object diagrams and genetic algorithm's tree cross over technique to generate test cases exhaustively. Then, the DFS algorithm is used to extract the test paths.

In summary, the UML based MBT methods either utilize a single diagram or multiple diagrams from the system model to generate test cases. The resulting test case trees are traversed using a search algorithm to identify the test paths. Some frameworks go a step further and prioritize test cases so that more emphasis can be given to most critical test cases. Similar to UML, the system representation in the HEFFR framework uses a combination of graphs to generate the system model. The similarity in the system representation and the benefits of MBT guided us towards using UML based test case generation as a basis for this research. However, they cannot be directly used because the system representation in UML is different than in HEFFR. As a result, we have studied the process the UML based approaches have taken and applied it in this research to automatically generate fault scenarios for the HEFFR framework.

4.2.3 Probability of Failure in Risk Assessment

Traditionally, engineers have relied on the probabilistic risk assessment to quantify the risk of failure [28]. Probabilistic risk assessment is the quantification of the risks due to hazards in terms of severity (how bad the hazard is) and occurrence (how likely it is to occur) [174]. Fault Tree Analysis (FTA) and Event Tree Analysis (ETA) are traditionally used to assess the risk of component failures during probabilistic risk assessment [175]. Both of these methods rely on principles from reliability engineering to calculate the probability of failure [28]. When a constant failure rate is assumed, an exponential probability distribution can be used as in Eq. 4.1 to calculate the probability of failure (P_f) of a component [176], where λ is the failure rate and t is the operation time.

$$P_f = 1 - e^{-\lambda t} \tag{4.1}$$

Human reliability assessment methods are used to quantify the human error probability in probabilistic risk assessment [28]. One common human reliability assessment method, THERP [34], uses event trees to model human errors and quantify them, giving minimal consideration for performance shaping factors. The Standard Plant Analysis Risk (SPAR-H) [38] method classifies tasks as action, diagnosis, or mixed based on them being physical, cognitive, or both, respectively. SPAR-H calculates human error probability using the task type, system operation status, task dependencies, and performance shaping factors. HEART [25] uses generic human error probabilities and performance shaping factors (called Error Producing Conditions (EPC)) to calculate human error probability.

This research uses a combination of HEART and SPAR-H to calculate HEP because these methods are easy to use, integrate well with actions in the action sequence graphs, apply to a variety of industries, and have the most potential to predict HEP with the minimal information available in early design. The probability of the component behavior modes is calculated using the exponential probability distribution described above, following processes previously outlined for early design reliability prediction [177].

4.2.4 Severity in Risk Assessment

In probabilistic risk assessment, it is necessary to assess the severity of failures so that risks can be prioritized and managed in proportion to their impact. Typically, FTA and ETA do not assess this severity of consequence(s), leaving the assessment up to the judgment of the designer. Severity is, however, assessed in detail in FMEA to prioritize faults and give details of the failure mechanisms and consequences [175]. In FMEA, each of these (as well as rate of detection) is rated on a 0-10 scale and multiplied into a risk priority number ($RPN = Severity * Occurrence * Detection$). However this approach has a number of limitations [178, 179]:

1. the ordinal scale for probabilities and severities distorts the relative impact of each since fault probabilities and costs often vary over orders of magnitude,

2. RPNs calculated by different project groups on different systems may not correspond to the relative risks of their subsystems because each number is subjective, and

3. there is no formal method to trade RPN for other desirable design attributes (e.g., to prescribe a risk-mitigating feature).

As a means of overcoming these limitations, expected cost has been presented as an alternative framework to design for risk [178–180]. When quantifying risk as an expected cost, the occurrence is quantified using the estimated number of times a failure scenario is to occur while the severity is quantified in terms of the cost incurred if that scenario occurs, according to:

$$C = \mathop{\mathbb{E}}_{s \in S}\{C(s)\} \approx \sum_{s \in S} n(s) * C(s) \tag{4.2}$$

where S is the set of fault scenarios, $n(s)$ is the lifetime number of occurrences for a scenario, and $C(s)$ is the modeled cost of a fault scenario. Expected cost can be used both for risk and resilience quantification for design optimization [181–183]. To integrate expected cost quantification with fault modeling tools, Ref. [184] considers three main costs: cost of failure, cost of repair, and cost of partial recovery [26]. Costs can also be added for risk using existing safety cost schedules (e.g., [185]),

Table 4.2: Functions, corresponding generic components, and their behavior modes

Functions	**Generic Components**	**Behavior Modes**
Import Liquid Guide Liquid Export Liquid	Valve	**Nominal On, Nominal Off,** **Failed Open, Failed Close,** **Stuck Open, Stuck Close**
Transfer Liquid	Pipe	Leak, Ruptured, Nominal
Store Liquid Supply Liquid	Tank	Nominal, Leak

provided one is at liberty to do so. This work adapts this quantification of expected cost to the HEFFR framework to enable designers to prioritize and make sense of hazards given by a large set of fault scenarios.

4.3 Proof of Concept Example: Hold-Up Tank

The same liquid tank design problem used in Chapter 3 is used to demonstrate the automated scenario generation approach and the risk quantification model. However, more context is added to the problem by converting it to a coolant tank to allow for severity prediction. The problem is to design a liquid cooling tank that can maintain its coolant level between a minimum and maximum threshold. The cooling tank is expected to maintain the temperature of a certain industrial machine that can explode if overheated (i.e., if the coolant level becomes too low). If the coolant level is above a certain level, the machine will cool down too much, resulting in severe damage. The coolant is a hazardous chemical which can cause health issues to human if exposed in large quantities. A human operator is expected to monitor the liquid level of the tank and shut off the incoming liquid if the water level is too high, and shut off the outgoing liquid if the water level is too low. This set-up is a simplified archetype of nuclear reactor and industrial plant operation, where maintaining optimal temperature is critical for both performance and safety.

The system representation of this problem, including the functional model, configuration flow graph, and action sequence graphs, is the same as what was displayed in Chapter 3. The system has eight functions and five components, two of which interact with the human. The operator will

interact with the valves to shut off the flow of liquid. The functions, corresponding components, and their behavior modes are shown in Table 4.2, where the human-induced behaviors are shown in **bold**. We use the function “Store Liquid" as the critical function because of its importance in maintaining the temperature of the equipment and its failure can impact safety.

4.4 Automated Fault Scenario Generation

The objective of this research is to develop an automated scenario generation technique that covers a majority of the component- and human-related fault conditions so that a comprehensive failure analysis can be conducted using HEFFR. In this section, the automated scenario generation technique is discussed and its application to the hold-up tank case study is demonstrated.

4.4.1 Methodology

We use the behavior model and the action classifications to generate fault conditions using transition functions and a modified Depth First Search (DFS). DFS is a tree or graph search algorithm that searches through a branch as far as possible before moving on to the next branch[186]. Each level of the branch is considered as a time step and each branch as an input scenario for HEFFR. A HEFFR analysis is done at every time step to check if the predefined critical functions are lost. If the functions are lost, the search stops and the path (branch) and the results (failures and propagation paths) are stored and the search moves to the next branch. The search continues until all branches are evaluated.

4.4.1.1 Transition function

The transition function is a set of rules that are used to create the child nodes for each mother node. For our application, these rules change the behavior mode (a state the component can be in, e.g.,

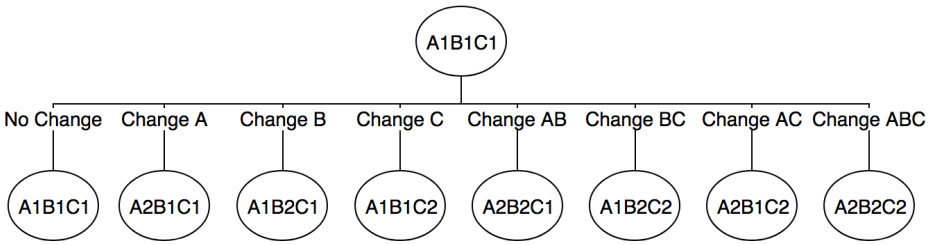


Figure 4.1: An example of an application of the transition function

for a pipe, it can be leaking, ruptured, or working as expected) of the components to induce faults. One of the rules makes no change to the mother node, creating a child node that is exactly the same as the mother. This is done to study the propagation of faulty behavior without introducing any further behavior modes. However, the number of time steps or number of consecutive child nodes that the "no change rule" is applied is limited by a user-defined number. The rest of the rules start with changing the behavior state of one component at a time until all behavior states for each component are applied. Then, the behavior modes are changed for two components at a time until all component combinations are done. This process increments until the behavior modes of all components are changed to make sure that all possible combinations are executed. For example, if the behavior modes of components A, B, and C are A1, A2, B1, B2, and C1, C2 respectively, the rules applied and the resulting branches for mother node A1B1C1 are shown in Fig. 4.1. In order to avoid explosion of scenarios, once a faulty behavior mode is introduced for a component, reverting back to a nominal state is not allowed. However, the transition rules do not stop child nodes from going to a previously analyzed faulty behavior state (i.e., if a pipe is leaking , it is allowed to go to clogged and back to leaking in future time steps. But it is not allowed to go back to nominal in the future). In reality, if a failure is present, usually, it does not go away unless it is repaired. However, one failure can propagate to another.In rare occasions, external influence can cause temporary faulty behaviors in components. Such malfunctions go away as the influencing factors resolve. For example, cold weather can cause fluid to freeze and clog fluid flow in a pipe until the weather improves. In such cases, designers are encouraged model those temporary behaviors as a subgroup of nominal behaviors to allow the algorithm to switch back and forth between the temporary and nominal behaviors.

When there are components that interact with the human, the rules are slightly modified. The behavior mode generation for the component still stays the same. However, the child nodes are allowed to go back and forth between nominal and faulty behavior for human-induced behavior modes, because these behaviors do not involve mechanical failures and they only depend on the human actions. When a non-human induced behavior is present, child nodes are not allowed to go to human-induced behaviors anymore. When a non-human-induced fault such as a mechanical failure is present in a component, a human's interaction with that component will not alter the mechanical failure or the system unless that component is repaired. Hence, considering human actions for such behavior modes does not add any value. Note that the action classifications (results of human actions) are not used to generate scenarios. Instead, the resulting component behavior modes are used to generate scenarios. The HEFFR framework assumes that the human can only interact with the system through its components. Thus, the human-induced behavior modes of the components cover the system-level effects of human error. Hence, we have chosen not to consider action classifications during scenario generation because this will only increase the possible combinations and not add any value in terms of understanding the system-level effects of human error.

Action classifications provide valuable information on how human-induced behaviors are produced. In order to give designers more details on what specific human actions contribute to human-induced behaviors, the algorithm does the following. When a human-induced behavior mode is present, all possible combinations of action classifications that can result in that specific behavior mode are generated using the action simulation and presented with the results of the overall simulation. Even when the behavior mode is nominal, combinations of action classifications are provided to the users to make sure that any human errors that fail to propagate to affect the component behavior do not go unnoticed. When generating the action classification combinations, the algorithm checks to see if the action classification combination is viable. For instance, one cannot see an invisible object. The algorithm checks for these relationships using the ASG and if combinations that violate these relationships are present, they are omitted. These steps make sure that the combinations presented to the user are as realistic as possible.

4.4.1.2 Critical Event Scenario Generation and Evaluation

Initialization: The number of consecutive time steps the “no change rule” can be applied is retrieved from the user. The users are advised to consider the maximum number of time steps required to enforce the loss of the critical functions when choosing the number of consecutive time steps. For instance, let us assume that the designers have modeled a hydraulic braking system in a way where a leak in the line will cause the fluid to empty in five time steps and the brake pad will wear down by 10% on each time step if there is a faulty behavior in the caliber. The number of time steps the “no change rule” should be applied should be ten instead of five because it is the maximum number of time steps it will take for the braking function to fail. Also, the maximum number of time steps that need to be simulated or the lowest level a branch can go to is also read from the user. These inputs make sure that the algorithm does not get stuck in a branch indefinitely. Next, the critical functions are read from the users. The behavior modes of all components and the action classifications of all actions are initialized to a nominal state. This will serve as the mother seed for the DFS. Also, the time step is initialized to zero.

Goal State: The algorithm continues through a branch until a goal state is achieved. The goal state in this case is the failure of all critical functions. When a goal state is achieved, the algorithm stores the critical event scenario input to HEFFR and the results from the HEFFR assessment.

Execution of the Transition Function: Two markers are used to track the application of the rules. They track the rules applied down and across a branch. When the transition rule is applied, these markers are updated. If the transition rule is the “no change rule", the algorithm checks if it has been applied for the maximum allowable consecutive applications. If it has, it moves on to the next rule. When a transition rule is applied HEFFR analysis is conducted to check if the critical functions have failed. If the critical functions have failed, the path of the behavior modes (i.e., branch) and the time steps are stored. The corresponding HEFFR results are also saved. Then, the search moves to the previous level and the next transition rule is applied. The above process is iterated until there are no more rules to be applied in level 1. Every time the search moves to a new branch, the last time step from that branch is picked up and incremented for the new child nodes.

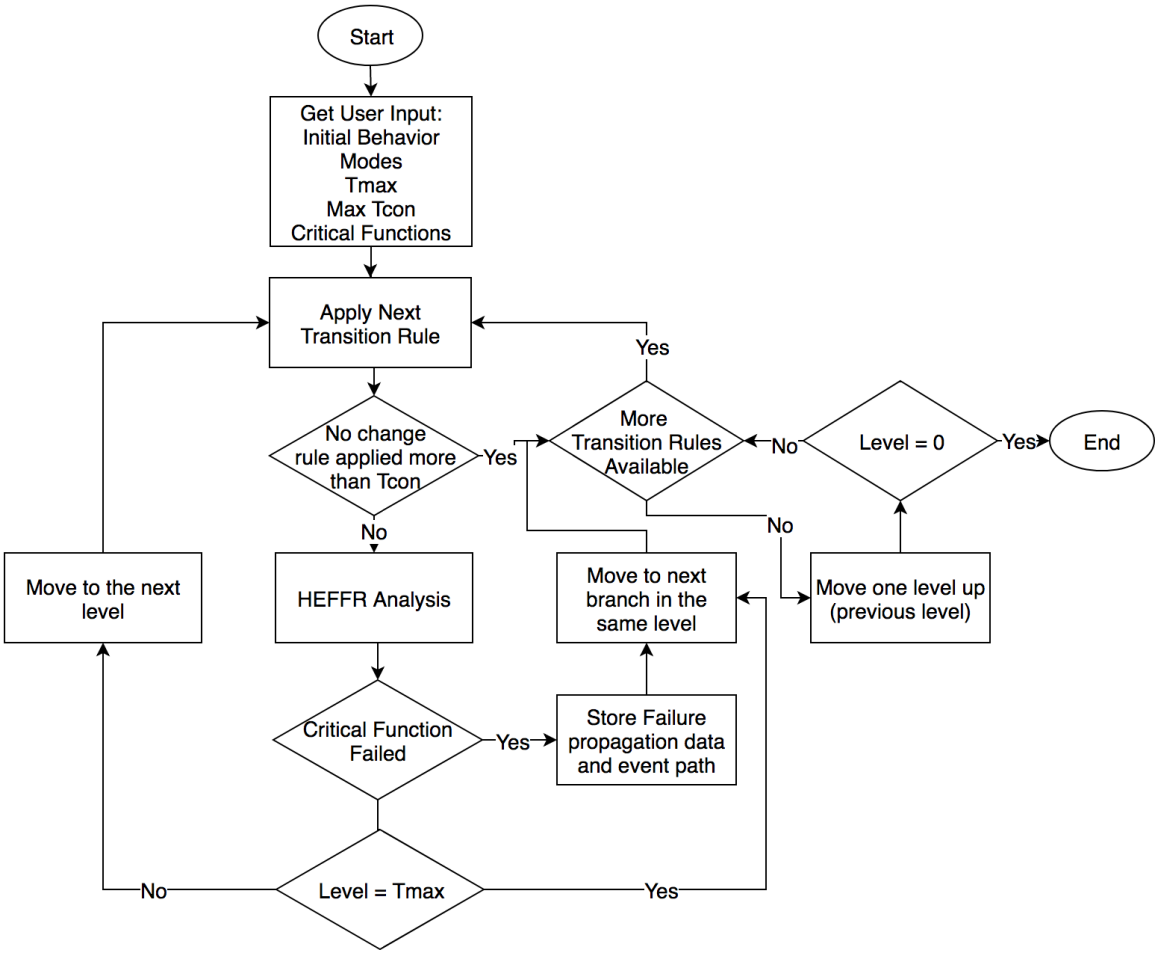


Figure 4.2: A high level flowchart of the automated scenario generation approach

If at least one of the critical functions has not failed, the child node becomes the new mother node and it is stored with its corresponding time step. Then, transition rules are applied to the new mother seed and the process above is repeated. If there are no rules to be applied at the current level, the search moves back one level and follows the above steps. Also, if the number of time steps reaches the maximum allowable time steps, the search returns to the previous level and the above steps are followed. This process is shown in a high level flowchart in Fig. 4.2.

4.4.1.3 Understanding the Results

The output of an execution will contain| the fault scenarios with corresponding time steps and the failures, human errors, and their propagation paths. We have chosen to leave the data in its raw form because it will give the designers the flexibility to analyze for what they are looking for specifically. For instance, if one is using this framework to identify the behavior modes that are most vulnerable in terms of having an effect on certain functions, they can define those functions as critical functions and look at what behavior modes were involved in the shortest paths that caused those functions to fail. Similarly, if one wants to compare alternative designs, they can compare the data from the runs for the alternatives to see which designs had the longest paths to failure on average. With the emergence of big data and advances in data science, there are a wide variety of tools to extract information. Hence, we present as much data as possible to designers so they can use such tools to extract information that is tailored to their needs.

Next, the results from the application of the automated scenario generation method to the hold up tank case study are presented. By executing this case study, we intend to explore if the algorithm is capable of creating effective use cases that cover a wide range of fault scenarios that involve component failures and human errors. Then, we study the results to see if the use cases were useful in identifying potential human errors, component failures, and their propagation paths by answering questions such as what are the fault scenarios that affect the critical functions of the system the fastest? We also intend to identify the behavior modes that have the highest chance of affecting the critical function. We do this by calculating the percentage of scenarios (of all scenarios identified to cause the critical function to fail) with each type of faulty behavior modes. Overall, we try to understand if the automated scenario generation method helps overcome the previously mentioned limitations of HEFFR.

4.4.2 Results

For this study, we chose the number of time steps a scenario should be allowed to propagate (number of times the "no change rule" should be applied) to two. This is because the behavior of the tank was set such that it would take two time steps to overflow or dry out depending on the flow of the liquid. Having more than two consecutive time steps is redundant since if the function Store Liquid were to fail due to a specific fault, it would in two time steps. Hence, analyzing the propagation of the same failure anymore does not add any further value. We chose the maximum number of time steps as five. Since the simulation continuously introduces faults at each time step, no single fault scenario can be repeated for more than two consecutive time steps, and the tank behavior drives the failure of function Store Liquid within two time steps, ideally, having up to four time steps would have revealed a majority of worse case fault scenario combinations. We chose five time steps so that we analyze a step further to uncover any unforeseen fault conditions. Note that in order to analyze five time steps (until t = 5), six time steps needs to be analyzed in total because the simulation starts at t = 0. Analyzing any further will introduce repetition of faults from previous sets of time steps. For instance, a scenario set that was present between time steps one and three may be repeated from time steps five to seven. Since the simulation is time-based, the number of potential scenarios is infinite if such repetitions are allowed. Hence, to avoid the explosion of scenarios, it is up to the user to choose the number of total time steps and number of time steps the "no change rule" can be applied wisely by considering critical functions and the behavior modes of the related components.

When creating faults for time steps, no temporary component failures were considered (i.e., once a non-human induced faulty behavior mode was introduced for a component, it was not allowed to go back to a nominal behavior mode. However, human induced behaviors were allowed to go back and forth between nominal and faulty behaviors). When generating action classification combinations that can result in human induced behaviors, the following rules were considered.

- The operators cannot detect a signal without being able to see it. Hence, when the action classification of the action See Water Level is "not visible," combinations with action Detect not equal to "not detected - failed" were omitted.

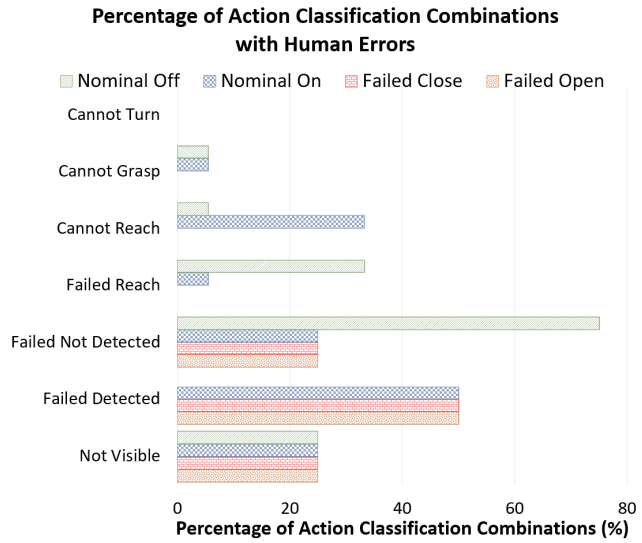


Figure 4.3: The percentage of action classification combinations with each human induced behavior

- One cannot grasp an object without reaching it first. So, when the action classification for the action Reach meant that the operator did not reach the valve, combinations with action Grasp not equal to ”no action" were omitted.
- One cannot turn a valve without grasping it. Hence, when the action classification of the action Grasp meant that the operator did not grasp the valve, the combinations with action Turn not equal to “no action" were omitted.

The simulation begins by taking the critical function as an input. Next, the behavior modes of all components are set to nominal - “nominal open” for the valves and “nominal” for the pipes and tank. The initial flow and the water level of the tank are then set to nominal. Then, the number of time steps a new node is allowed to propagate (“no change rule” is applied) and the total number of time steps are set. Once these inputs are read, the algorithm begins to evaluate critical event scenarios automatically. The execution took around 28 minutes on a personal laptop (IntelCore i5, 2.9 GHz speed, and 16 GB RAM) which is reasonable considering the amount of information that can be extracted. In total, around 15 million event scenarios resulted in the function Store Liquid failing in t=5. Only two scenarios were found to have purely human induced faulty behaviors whereas 163,204

scenarios had purely non-human induced behaviors. The rest of the scenarios had a combination of both non-human and human induced faulty behaviors.

Out of the 1,824 possible action classification combinations only 152 were generated. The rest were omitted based on the rules listed above (because they were not realistic). Out of the generated action classification combinations, 4 each resulted in human induced behaviors "failed open" and "failed close." There were seventy-two combinations each for "nominal on" and "nominal off." Additionally, 86% of the action classification combinations that contributed towards "nominal on" and "nominal off" had underlying human errors (actions in faulty classifications, for example "cannot turn" - when an attempt to turn is made and cannot be physically achieved, but not due to a component failure) for at least one action meaning that these errors did not propagate to affect the system. However, they must be considered when making design decisions because they may affect the system as the design evolves. Seventy-five percent of all combinations that led to "nominal off" had the human error "failed not detected." Similarly, detect related human errors were prevalent across all behavior modes (50% "failed detection" in "failed open," "failed close," and "nominal on") followed by reach, grasp, and vision-related errors. The details of this analysis are shown in Fig. 4.3, where the percentages are calculated by considering the number of times an action classification was present in the action classification combinations that could result in a specific human induced behavior.

The shortest path for failure was at four time steps (t=3). There were 10,459 event scenarios that led to the failure of the Store Liquid function in four time steps. All of the event scenarios had the failure type "leak" for the component tank at least once. This is expected because the function Store Liquid is directly fulfilled by the tank. A leaking outlet pipe and a clogged inlet pipe were other commonly occurring failures (79% and 83% respectively). "failed open" was most common for the outlet valve (78%) and "failed close" was most common for the inlet valve (78%) among human induced behavior modes. The prevalence of the behaviors "leak" tank, "clogged" inlet pipe, and "failed open" outlet valve indicate that a majority, if not all of these event scenarios resulted in a tank dry out. The detailed analysis of the presence of each faulty behavior mode in the event scenarios with the shortest path to critical function failure are shown in Fig. 4.4.

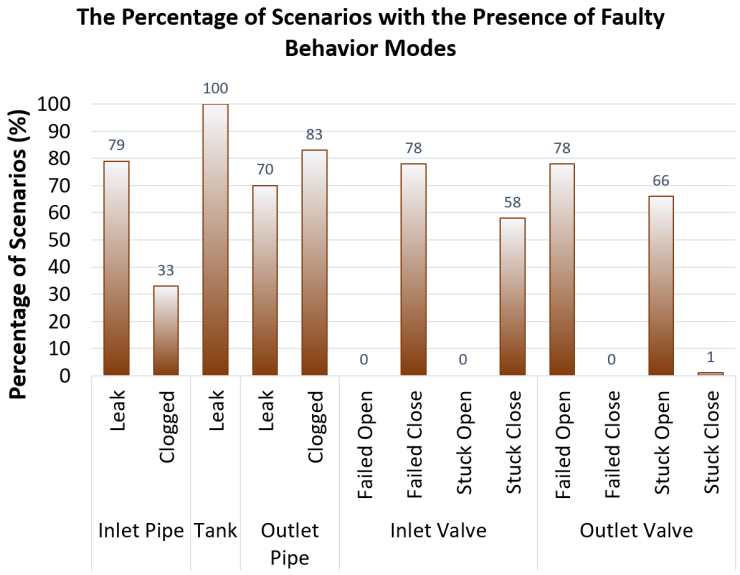


Figure 4.4: The percentage of total event scenarios with each type of faulty behavior mode

Using this data, designers may make design decisions to avoid the leak in the tank and the outlet pipe by trying different materials, adjusting wall thickness, or recommending additional testing of these components. Similarly, they may choose to add sensors to detect clogs in the outlet pipe or check the quality of the liquid to make sure that there is no residue build-up. For the human induced behaviors, detect related errors are most common followed by reach, grasp, and vision-related errors. To prevent detect related errors, designers may chose to make changes to the system by adding redundant signals, or making signals more salient. On the other hand, they may suggest training to improve the operators ability to detect signals. Designers may choose to conduct DHM analyses such as reach analysis, percent vision obscuration, etc. to identify ways to mitigate non-cognitive human errors. Depending on the workspace design, they may also choose to perform ergonomic assessments such as comfort, lower back compression force, and biomechanics. They may apply human factors engineering guidelines, suggest training, or device operational procedures to mitigate all types of human errors. In contrast, they may not resort to design decision yet and further analyze the data. They may choose to find out how the behavior of other functions contribute to the failure of the tank or repeat the execution with different critical functions with different input conditions. Either way,

when design changes are made, they may update the system model and iterate through this process until a satisfactory design is derived. The designers may set the criteria for a satisfactory design as they see fit. Note the HEFFR framework, being an early design stage tool, can only model changes the system functions, components, and human action sequences. It cannot represent changes to component parameters (such as thickness, material properties, etc,) and human action parameters (change in anthropocentric, environmental conditions, etc.), which usually come to light during later design stages. Hence, the analysis should only be iterated when such changes are made to the design.

4.4.3 Discussion

The hold-up tank example shows how the automated scenario generation can be used with HEFFR to identify potential functional failures, human errors, and their propagation paths early in design. The presence of both human induced and non-human induced behaviors in the scenarios indicate that the automatically generated scenarios included fault conditions involving both humans and components. The results showed the importance of avoiding behavior modes such as “leak” tank and “clogged” input valve to mitigate the potential loss of the function Store Liquid. Among human actions, detection related human errors were most prevalent stressing the need for mitigating such errors. Additionally, human errors were present even when the components were in nominal behavior modes. Even though these human errors did not propagate to affect the system under the circumstances the system was analyzed, it is important to consider them when making design decisions because they may be exposed in different circumstances.

The above results show the ability of the algorithm to generate a broad spectrum of fault scenarios involving both components and humans that can violate the critical function of the system. Only a small percentage of generated scenarios had purely human induced behaviors or non-human induced behaviors, meaning that if human errors or component failures were evaluated in isolation, a majority of potential fault inducing conditions would have been missed. This shows the importance of assessing the effects of both component failures and human errors acting in combination to better understand

potential risk and promote appropriate mitigation strategies. The data analysis presented in this paper is minimal when compared with the information that can be extracted from the data. Only t = 3 was analyzed in this example. Further analysis can be done at t = 4 or t = 5. Such an analysis can give important information on how the system will operate under more regular and less severe fault conditions. The ability to identify human errors that do not propagate to affect the system is another plus since most system failures occur through failed to detect, compounding failures. Overall, the proposed approach allows for a more comprehensive failure reasoning through input scenarios involving the fallibilities of both humans and components. The results can be used to conduct various types of analyses, depending on the needs and requirements of the designers.

The performance related measures such as the execution time and the number of scenarios that caused the critical function to fail of this case study do not necessarily reflect the overall performance of the algorithm. Considering the amount of information that can be extracted and how useful they can be to design a safer, more reliable system, the execution time for this problem was acceptable. However, the execution times surrounding more complex engineered systems and if they outweigh the benefit of the information received is yet to be seen. A large number of event sequences were identified to violate the critical function. We may never know if these numbers mean that the scenarios included all of the possible combinations of human errors and component failures since there is no such data to compare the results with. However, one can be assured that a majority of such fallibilities were covered because the algorithm evaluated all possible combinations of behavior modes at least once and the large volume of results mean that different combinations of these behavior modes was evaluated.

The number of total time steps and the number of time steps where the same scenario can be executed consecutively play a significant role in the total number of scenarios evaluated. In fact, the number of scenarios increases exponentially with each time step. Since the simulation is time-based, there is nothing that limits the number of scenarios except for the time. While having a lot of data is important to be able to extract a wide variety of information, repetitive data in large volumes can make this process slow, resource intensive, and at worse impossible. Designers are encouraged to

consider the critical functions and the component behavior modes that can induce a failure carefully to minimize the time-related variables. For example, in the liquid tank case study, the behavior mode of the tank dictated that it will take two time steps for the tank to dry out or overflow from a nominal level. That determined the total number of consecutive steps as two and total time steps as five. If the behavior mode of the valve only required one time step for a failure to occur and a designer was evaluating the function related to the valve as the critical function, he or she may choose one or two for consecutive time steps and two or three for total time steps.

Another major contributor to the breadth of the search tree is the behavior modes of individual components. As the behavior modes increase, the number of combinations of potential scenarios increases exponentially, which in turn increases the total number of available scenarios. Hence, users are encouraged to carefully consider behavior modes and avoid any redundancy. Based on the cases presented above, one can argue that the proposed scenario generation method is vulnerable to data explosion (i.e., too many scenarios being created). However, these vulnerabilities encourage designers to think about the behaviors of the components that will be part of the system they are designing early in the design process, which can lead to well thought out designs. Additionally, the explosion of the scenarios can be avoided if the contributing variables are handled carefully.

Another way to avoid data explosion is to take a systems-of-systems approach into the analysis. Complex systems can be broken down to less complex subsystems. The proposed approach can be applied to these subsystems to identify the potential vulnerabilities of them. The resulting data can be used to construct the whole complex engineered system in which the black box functions of each subsystem will build the functional model, the subsystems themselves will be components in the configuration flow graph, and the subsystems that interact with human will have action sequence graphs. Then, the proposed automated scenario generation can be applied to the new system model. Such an approach will help designers pay attention to individual components in more detail while making sure that the overall system vulnerabilities and human fallibilities are addressed. Overall, the proposed automatic test case generation technique overcomes some of the shortcomings of the HEFFR framework. The scenarios generated, cover a wide variety of failure conditions involving

both humans and components. This approach also allows the analysis of a large number of scenarios at the same time. The proposed automated scenario generation technique may be prone to data explosion. However, data explosion can be avoided and the path to mitigation will help designers come up with more thought out concepts.

4.5 Quantifying Risk

As discussed in section 4.4.3, the automated scenario generation technique can help designers to mine important information such as the shortest event sequence to failure, the likelihood of a particular behavior mode being present during a failure, and the possibility of specific faulty human actions being present in human induced behaviors. However, such metrics do not quantify risk in terms of likelihood and severity. Without risk quantification, designers will have to treat all scenarios equally and will not be able to prioritize fault scenarios to implement design solutions. When a large number of scenarios are present, it becomes infeasible to give equal consideration to all scenarios. This section overcomes this limitation by introducing a probability and cost model to the HEFFR framework to quantify the risks of human and component-induced failures.

4.5.1 Methodology

The objective of this research is to aid designers identify and prioritize high severity fault scenarios that result from the interactions between component failures and human errors (in addition to the scenarios that result from them acting independently) during the conceptual design stage. We use two metrics–the likelihood of failure, and expected cost–to achieve this goal. This is done by processing the output of HEFFR to calculate the above metrics using cost and probability models. The following definitions will be used for the terms event and scenario for the rest of this dissertation:

- Event: the behavior state of components and the human action classification states in a timestep
- Scenario: a collection of events

Table 4.3: HEFFR sample result: Fault scenario input and resulting functional failures

Critical Event Scenario (HEFFR Input)				**Functional Failure (HEFFR Output)**		
t	**Generic Component (GC) 1**	**GC2**	**GC3**	**Function (F) 1**	**F2**	**F3**
0	Nominal Behavior Mode (NBM) 1	NBM1	NBM1	Nominal (N)	N	N
1	Faulty Behavior Mode (FBM) 1	FBM1	NBM2	Degraded (D)	D	N
2	FBM1	FBM1	NBM2	D	D	D
3	Human Induced Nominal Behavior (HINB) 1	FBM1	NBM2	N	D	N
4	Human Induced Faulty Behavior (HIFB) 1	FBM2	FBM2	Failed/Lost (L)	D	L

Table 4.4: HEFFR sample result: Human action classification combinations and resulting human induced behaviors of component 1

Human Actions Inputs			**Resulting Human Induced**
Action 1	**A2**	**A3**	**Component Behavior**
Nominal Action Classification (NAC) 1	NAC1	NAC1	Human Induced Nominal Behavior (HINB) 1
Faulty Action Classification (FAC) 1	NAC1	NAC1	HINB 1
FAC2	FAC1	NAC2	Human Induced Faulty Behavior (HIFB) 1
NAC2	FAC1	FAC1	HIFB1
FAC 1	FAC2	FAC2	HIFB1

A sample output for a HEFFR assessment of one scenario is presented in Tables 4.3 and 4.4. Table 4.3 shows the critical event input at each time step and the resulting health of functions. Table 4.4 shows the human action classification combinations and the resulting human induced behaviors of a component.

4.5.1.1 Calculating the Cost of a Scenario and the Expected Cost of the System

The costs of a scenario come from disruptions to safety and performance, and required repairs [26, 184]. To quantify these costs, we use Eq. 4.3, where C_s is the cost of a scenario, C_f is the immediate cost (e.g., due to safety impacts), C_p is the performance cost (e.g., due to lost functionality), t_r is the time to recover, C_r is the cost of repair, NF is the functions in faulty states, and FC is the components in faulty behavior mode. For C_f and C_p, all functions that are not in a nominal state are

considered. For C_r, all components that are in a non-human induced behavior mode is considered because human induced modes do not constitute damage, since they can be changed back to nominal by the operators. Depending on if the components are repaired in parallel or series, t_r will be equal to the recovery time of the behavior mode with the longest recovery time or the sum of the recovery times of all faulty behavior modes in the scenario.

$$C_s = \sum_{i \in NF} C_{f,i} + (\sum_{i \in NF} C_{p,i}) \times t_r + \sum_{b \in FC} C_{r,b} \tag{4.3}$$

To adapt this cost model to the system of interest, immediate cost and the cost of lost performance must be specified for the “lost" and “degraded" states of each function, as well as the repair cost and recovery time for each behavior mode of each component, which may be estimates based on historic data. Any safety costs can be incorporated using cost schedules applicable to the industry (e.g., [185]). The cost of a scenario is calculated based on the functional status and the behavior modes of the components in the final time step (i.e., t = 4 in Table 4.3) of a scenario. We use Eq. 4.4 to calculate the expected cost of failure of the system C_F, where T is the life-cycle time, λ_s is the failure rate of the scenario, C_s is the cost of a scenario calculated in Eq. 4.3, and F is a set of failures. Since HEFFR output scenarios are only those that cause the critical functions to fail, this cost calculation is only tabulated for those failures, which may be an incomplete set. Since the probability of failure is defined as in Eq. 4.1, the term $T\lambda_s$ in Eq. 4.4 is calculated using Eq. 4.5, where P_s is the probability of the failure scenario.

$$C_F = \sum_{S \in F} T\lambda_s C_s \tag{4.4}$$

$$T\lambda_s = -ln(1 - P_s) \tag{4.5}$$

4.5.1.2 Calculating the Likelihood of a Scenario

In this model, the behaviors of components in a time step and the events instantiated between time steps are considered independent. This assumption is made for simplicity and because the HEFFR simulation accounts for failures through the functional health of the system and the cascading effects of failures through the automated scenario generation. That is, when a new event is introduced to a scenario, no further events are introduced for a user defined number of time steps and the simulation allows the failures to propagate at the functional level. Hence, the interdependent failures will be assessed through their propagation at the functional level, so each event in a scenario can be thought of as an independent initiating event. Moreover, when an event has multiple faulty component behaviors present, each faulty behavior is considered to be independent of the others. Since the scenario generation considers all possible combinations of behaviors, cascading effects of every faulty behavior occurring alone or in tandem with others will be evaluated during the simulation. This is done mainly for simplicity; while more detailed models, such as Markov Chain Monte Carlo models or Bayesian graphs, represent events or behaviors as probabilistically dependent [187], these methods rely on transition probabilities which may be difficult to specify in the early design stages.

Calculating the Probability of Non-human Induced Behavior Modes: The probability P_f of a component operating in a faulty behavior mode is calculated using Eq. 4.1, assuming a constant failure rate (λ) and an exponential probability distribution. For t in Eq. 4.1, the expected product lifetime should be used. The probability of a component operating in a nominal behavior mode (P_n) is determined using Eq. 4.6. We recommend using Nonelectronic Parts Reliability Data (NPRD) and Electronic Parts Reliability Data (EPRD) to source component failure rates. These documents are created through a rigorous data collection process where historic failure events, maintenance records, and published data are used to present component failure rates [188, 189]. NPRD and EPRD consider a component to be failed when a part is repaired/replaced, and the failure symptoms were no longer present [188, 189], meaning that human induced behavior modes are not considered during the failure rate calculations.

$$P_n = 1 - P_f \tag{4.6}$$

Failure Mode/Mechanism Distributions (FMD) publishes the probability of failure modes and mechanisms of components, given that there is a failure [190]. The data is sourced and scrutinized similar to NPRD and EPRD [190]. When a component is in a nominal behavior state, the probability of the current behavior of a component P_c is equal to P_n. When a component goes to a faulty behavior mode from a nominal state, data from FMD can be used to calculate the probability of a specific faulty behavior mode P_{fb} using Eq. 4.7, where P_{fm} is the probability of a faulty behavior mode given that a failure is present. In that case, P_c is equal to P_{fb}. When a failure mode is already present for a component, P_c is equal to P_{fm} of the current behavior mode, because a component that is in a non-human induced behavior mode needs to be repaired to go back to a nominal state, making the probability of it returning to nominal 0. A failure mechanism is the process that caused the failure, whereas a failure mode is the effect of the failure observed [190]. In HEFFR, the behavior modes of components are similar to failure modes. Hence, only the failure mode probabilities need to be sourced from FMD. Since FMD does not distinguish between the probabilities of failure modes and mechanisms [191], failure mode probabilities need to be normalized to omit the mechanism probability distributions. For instance, if a component has two modes and a single mechanism, each mode has a distribution probability of 0.2 and 0.3, and the mechanism has a distribution probability of 0.5, the normalized probabilities of the modes will be 0.4 and 0.6, respectively.

$$P_{fb} = P_f.P_{fm} \tag{4.7}$$

Calculating the Probability of Human Induced Behavior Modes: We use a combination of SPAR-H [38] and HEART [25] methodologies to calculate the probabilities of human induced behaviors. In HEEFR, ASGs are used to track the human actions that need to be performed to interact with a component. Since the tasks in HEART are at a higher level (e.g., reduce speed) than actions (e.g., grasp object) in ASGs, a direct comparison between the generic tasks in HEART and actions in ASGs may be confusing. Hence, we propose the partial use of SPAR-H to assign human actions to generic tasks. As in SPAR-H, we propose that designers designate each human action in an ASG as “action,” “diagnosis,” or “mixed” if they are physical, cognitive, or both,

respectively. For instance, the action Reach will be designated “action” since it is physical, whereas action Detect will be designated “diagnosis” for being cognitive. The authors of SPAR-H have provided the HEART generic tasks that are comparable with these designations, where generic tasks D and F are comparable with designation “action” and generic tasks A-H, and M are comparable with designations “diagnosis" and “mixed.” Designers can use these comparisons to assign generic tasks to human actions in the ASGs (generic task descriptions can be found in Ref. [25]). When assigning generic tasks to human actions each action should be assigned one generic task (generic tasks A-H and M are listed in Appendix A).

The next step is to assign Error Producing Conditions (EPC) for each ASG in the system representation. In total, there are 38 EPCs that can be assigned. A list of the HEART generic tasks and EPCs are provided in Appendix A. Note that the EPCs are evaluated for whole ASGs and not for individual actions, because the EPCs in HEART are more relevant at the task level and not at specific action levels. In practice, HEART assessment requires the assignment of generic tasks at the task level also, but doing so will not enable one to identify the actions that contribute to a task failure. Hence, we have proposed the application of generic tasks to specific actions in the ASGs to calculate the probability of individual actions failing. Some of the generic actions already incorporate EPCs. The authors of HEART recommend omitting EPCs that are already incorporated in generic tasks to avoid overestimation. The proportion of effects of an EPC must be evaluated for each action in the ASG because the effect of these factors on each action may vary depending on the action performed. In summary, the probability of an action in the ASG failing can be calculated using Eq. 4.8, where P_{hf} is the probability of a human action failing, P_g is the generic task probability from HEART, EPC is the error producing condition factor, and x (between 0 and 1) is the proportion of effect of EPC. Then, the probability of a nominal human action (P_{hn}) can be calculated using Eq. 4.9.

$$P_{hf} = P_g \times \prod_i ((EPC_i - 1) \times x_i + 1) \tag{4.8}$$

$$P_{hn} = 1 - P_{hf} \tag{4.9}$$

When a human does not attempt to perform an action, identifying if that action is in a nominal or faulty state depends on the context of the overall system. For instance, if operators detect some signal which requires them to reach a valve, grasp it, and turn it off, and they do not attempt to do so, these actions will be classified as faulty. However, if there was no signal and they are not expected to turn off the valve, and they make no attempt, the actions will be classified as nominal. Hence, we consider the human induced behavior in the previous time step and the current time step to determine if a human action is in a nominal or faulty classification when no attempt to perform an action is made. A human action can have multiple nominal and faulty classifications. However, the calculations only identify the probability of a nominal or faulty action classification: they do not go into detail to calculate the probabilities of specific classifications, since there is no direct method to calculate such probabilities like there is for component behavior modes. Hence, to identify the influence of specific action classifications, data mining approaches (e.g., those in section 4.4) will need to be used.

Once the probabilities of each action in the ASG is assigned, the probability of the resulting human induced behavior can be calculated using Eq. 4.10. Multiple action classification combinations can result in the same human induced behaviors. Hence, a union of all these action classification combination probabilities is taken to calculate the actual probability of a human induced behavior P_h. For instance, if a human induced behavior has two action classification combinations and their probabilities calculated using Eq. 4.10 is equal to $P_{h'_1}$ and $P_{h'_2}$, the actual probability of the human induced behavior is calculated using Eq. 4.11. Since the human induced faults are not considered in the failure rate calculations, the probability of nominal component behavior (P_n) incorporates the human induced behaviors. Hence, when a human induced behavior is present in a component, the probability of the current behavior P_c is calculated using Eq. 4.12.

$$P_{h'} = \prod_i P_{hf,i} . \prod_j P_{hn,j} \tag{4.10}$$

$$P_h = P_{h'_1} + P_{h'_2} - P_{h'_1} . P_{h'_2} \tag{4.11}$$

$$P_c = P_h.P_n \tag{4.12}$$

Calculating the Probability of an Event and a Scenario: The next step is to calculate the probability of an event P_e using Eq. 4.13, where i is all components. Then, for every time step j, where the event is not equal to the event in the previous time step, the probability of a scenario is calculated using Eq. 4.14. When an event is allowed to propagate, no new events are introduced in the following time step. Hence, such time steps are omitted in the probability of scenario calculation. Note that since the simulation is time-based, each time step represents a discrete change in system state, and the simulation runs until a critical function has failed or a maximum number of time steps are reached, the total number of time steps need to be chosen to minimize event repetition. More details on how to choose the total number of time steps are provided in section 4.4. If not the simulation may become computationally expensive. In summary, the simulation takes inputs for costs, failure rates, system life cycle time, human generic tasks, EPCs, and EPC proportion effect factors in addition to the HEFFR automated scenario generation inputs. After the simulation, the cost of a scenario, expected cost of failure of the system, probability of a scenario occurring, and probabilities of action classification combinations are generated along with the outputs from the HEFFR automated scenario generation simulation.

$$P_e = \prod_i P_{c,i} \tag{4.13}$$

$$P_s = \prod_j P_{e,j} \tag{4.14}$$

4.5.1.3 Understanding the Results

The execution of the simulation yields a list of fault scenarios that result in the critical function failing and their probabilities, costs, and expected costs. We have not added any data synthesis as part of this work. Instead, we provide as much data as possible so designers can extract information tailored to the requirements and challenges of the system they are designing. The goal of this approach

is to not give exact probabilities for action/task or component failures, but to provide estimates that are reliable enough to study the relative impact of faults during conceptualization. Providing detailed probabilistic models for failures requires very specific information relating to the system. To comprehensively quantify human error probabilities, details on actual tasks, the environment where the task is performed, and the operator need to be considered. Since this information is not readily available early in design, estimating the corresponding probabilities during early design stages is difficult and subject to uncertainty. Thus, we focus on providing designers with an appropriate level of model fidelity to identify and prioritize risks early, without making the analysis too detailed.

Next, the results from applying the risk quantification model to the hold-up tank case study is presented. In this case study, we demonstrate the use of expected cost modeling to quantify risk in an HEFFR simulation. We then explore the results to see if the proposed method is capable of giving insight to designers about potential worst-case fault scenarios that cause the critical function to fail by answering the following questions. Can we prioritize fault scenarios in terms of severity and likelihood? Are there any fault scenarios that can be discarded? What are the worst-case component behavior modes? What is the contribution of specific human actions to failures? In summary, we try to understand if the proposed risk metrics calculations can help designers quantify the risk of component failures and human errors acting in combination and identify and prioritize worst-case fault scenarios to inform risk mitigation.

4.5.2 Results

First, the actions were designated as “action,” “diagnosis,” or “mixed.” Then, the generic tasks were chosen based on the nature of the actions and their ability to match with generic task descriptions. For example, the action Grasp was assigned generic task D because it is a simple task requiring minimal attention. Six EPCs were identified for each ASG based on the system model and the expected human-system interactions. The EPCs related to the operator (e.g., operator experience) were not considered since no such information is available in the design problem. We assigned the

proportion of effects of each EPC for each action based on the action performed, the component the action will be performed on, how the action will be performed, and the conditions surrounding performing the action. For instance, for the EPC "no clear direct and time confirmation of an intended action from the portion of the system over which control is to be exerted," the action Detect was assigned a proportion of effect 0.1 for both inlet and outlet valves because the system model did not include any means of confirmation for the detection of signal. However, if the operator fails to detect the signal, it will only affect the system if they acted upon it (through action Turn, a different action), making the proportion of effect of the EPC for the action Detect low.

We selected the failure modes of components from NPRD-95 [192], assuming the Ground Fixed operating environment when rates were available. If they were not, failure rates from other ground environments were chosen depending on their applicability for this case study. The total lifecycle time was chosen as two years, given that the system will be in constant operation. The failure mode distributions of the components were selected from FMD-97 [193]. The repair costs of the components were estimated based on the cost of part replacement and diagnosis. The recovery times included the time to repair components and time to clean up any resulting spills. The performance costs of each function considered the impact of the function being degraded or lost on overall system performance. The immediate costs of each function were estimated considering the chemical exposure, safety, and necessary cleanup if there was a coolant spill. Assigned values relating to human actions, components, and costs are available in Appendix B.

The simulation begins by taking the critical function (Store Liquid failing), initial component behavior modes (nominal for all components), initial liquid flow rate (nominal), initial tank coolant level (nominal), the maximum number of time steps (4), and the number of time steps a failure event is allowed to propagate (2) as inputs. Next, the inputs for likelihood of occurrence and expected cost calculations are read (the input values are listed in Appendix B). In total, around one million scenarios that could cause critical function failure were generated. The total expected cost of the system was found to be around one million dollars. The highest failure cost was around 52 million dollars, and the maximum likelihood of occurrence was around 3.5×10^{-3}. The lowest-likelihood

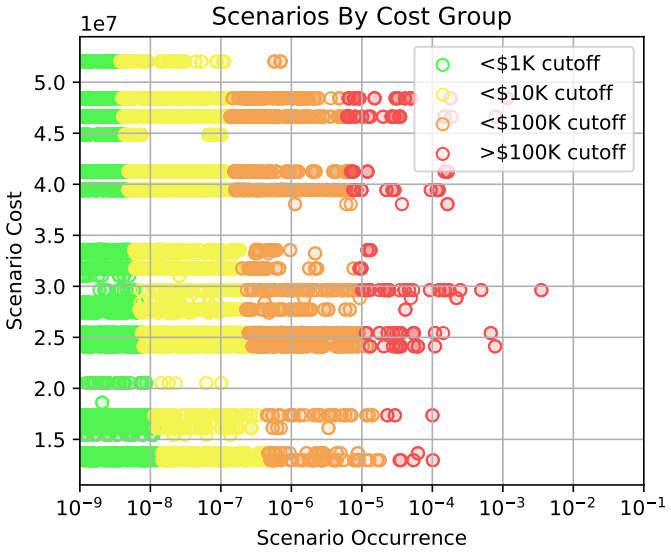


Figure 4.5: Cost groups of fault scenarios

scenario had a probability of around 2.8×10^{-24}, and the minimum scenario cost was around 13 million dollars. A majority of the scenarios modeled had low probabilities and thus low expected costs. One reason for the low probabilities is the independence assumption used in the probability model, where the probability of every independent behavior or event is multiplied with the others. Another reason for this is because adverse events are rare by definition. As a result, it may be expedient to put the scenarios in groups so that the high-cost scenarios are given priority.

Figure 4.5 shows a set of priority groups for the scenarios by setting a cut-off for the expected cost of scenarios. As shown, the *cumulative* expected cost of the scenarios in the green is below 1,000 dollars. As a result, the designer may choose to ignore them. The cumulative expected cost of all scenarios in yellow is less than 10,000 dollars, and thus *may be* worth considering as a group. The high-impact scenarios are labeled in orange and red, with the highest impact scenarios in red. Based on these cut-offs, these scenarios should be given individual attention to mitigate hazards effectively. For instance, one of these worst-case scenarios caused the critical function to fail in three time steps. In the first time step, the tank is in a faulty behavior mode ("leak"). In the next time step, no further failures are present (failure from the last time step is allowed to propagate). Finally, the

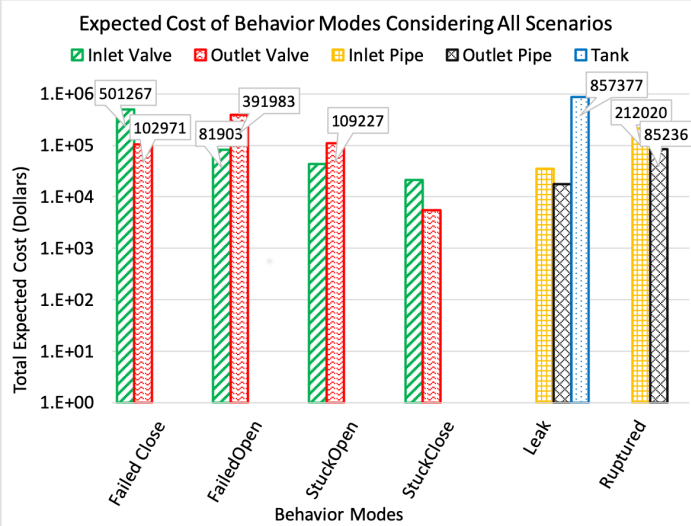


Figure 4.6: Expected cost of behavior modes

outlet valve goes into a human induced faulty behavior ("failed open") while the tank is still leaking. The benefit of this cost model is it identifies high-cost, high-probability scenarios like this and gives them priority over less likely, less costly scenarios.

The impacts of each fault mode can be assessed by calculating the respective cumulative expected cost of scenarios for each (Fig. 4.6). The behavior modes "leak" for the tank and "ruptured" for the inlet pipe have the highest expected cost among the non-human induced faulty behaviors. Designers may mitigate these risks by selecting components with lower failure rates, adding redundancies, including advanced failure detection mechanisms, performing tests to understand the failure mechanisms, and minimizing the chemical exposure when a failure occurs. As shown, the human induced faulty behaviors for inlet and outlet valves ("failed open" and "failed close") have a high expected cost. Hence, further assessment is needed to understand the specific human action combinations that contribute to the faulty human induced behaviors. Figure 4.7 shows the maximum reduction of probability by eliminating action classification combinations. For example, for the inlet valve, the probability of faulty behaviors can be reduced by 80%, if the top 45 of the total 112 action classification combinations are eliminated. While one cannot *eliminate* action classification combinations,

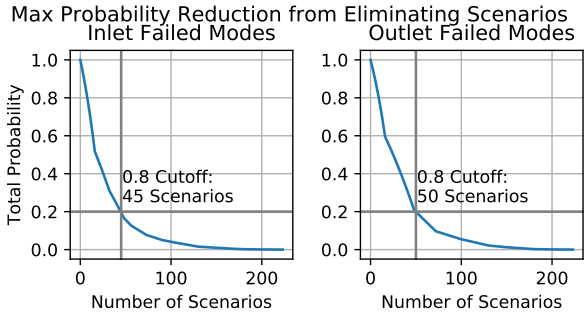


Figure 4.7: Maximum probability reduction from human action combination elimination

this plot shows that the human failure probability can be reduced significantly by focusing on a small subset of combinations. The human action failure probabilities of these combinations can be reduced by changing the system design and operational setting.

Figure 4.8 shows the number of times faulty action classifications are present among the scenarios that can reduce the likelihood of behavior modes "failed open" and "failed close" of both valves by 80% each. As shown, faulty action classifications were only present for actions Detect, Reach, and Turn. Also, not all faulty action classifications of these actions were present (e.g., "cannot reach" for action Reach). Cognitive errors (detection related and when actions are not attempted) were more prevalent when compared with non-cognitive errors ("cannot turn"). One way to reduce the likelihood of the faulty action classifications occurring is to reduce the effect of EPCs through the design of the system. For instance, designers may include action feedback mechanisms to eliminate EPC-14. For cognitive errors, the designers may suggest training or operating procedures to improve operator situation awareness as a means of mitigating them. They may also follow human factors engineering guidelines to improve the design to support error mitigation. For the non-cognitive human errors, they may use Digital Human Modeling to visualize the interaction and perform further ergonomic assessments.

One of the major limitations of using an expected cost model to prioritize fault scenarios is that the input information (rates and costs) may be low-fidelity. In this situation, it is important to understand how changes in the model inputs variables affect the expected cost of scenarios and thus the results of the analysis. To consider this uncertainty, we performed a Sobol [194] sensitivity

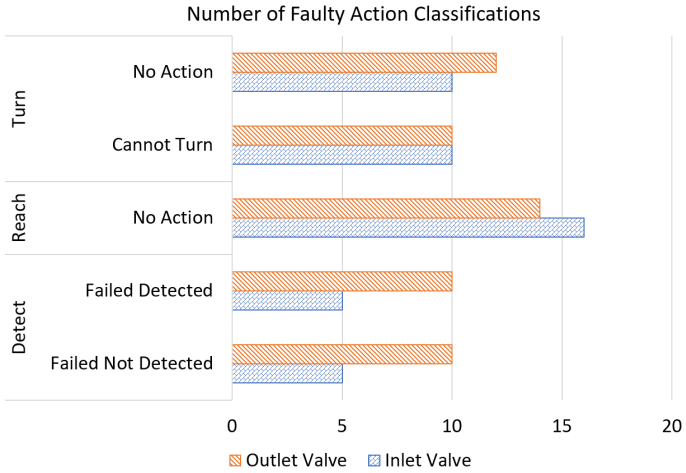


Figure 4.8: The number of faulty action states

analysis with 10,000 samples (using Saltelli [195] sampling 560,000 model inputs in total) for the hold-up tank study. Performance cost, repair cost, repair time, immediate cost, the proportion of effects for error producing conditions, and failure rates related variables (54 in total) were considered to have uncertainty. When assigning uncertainty ranges, ±20% of the original values were used to assign minimums and maximums for all variables except for the proportion of effects for error producing conditions related variables. For the proportion of effects for error producing conditions related variables ±0.2 of the original values were used to represent the uncertainty better. Since variables considered in the expected cost calculation vary scenario-to-scenario, we randomly chose 100 scenarios to perform the sensitivity assessments. The average first order and total sensitivity indexes were calculated to understand the sensitivity of the model for each of the input variables with uncertainty. The first order sensitivity index indicates the effect of individual input variables, whereas the total sensitivity index indicates the effect of individual variables and the effect of all interactions.

Figure 4.9 shows the averages of sensitivity index values of variable groups. Failure rates and the proportion of effects of EPC factors had the highest first order and total sensitivity indexes, followed by repair time-related variables. The first order and total sensitivity indexes for the other variable groups were negligible. Among the failure rates, the failure rate of the pipe had the highest first

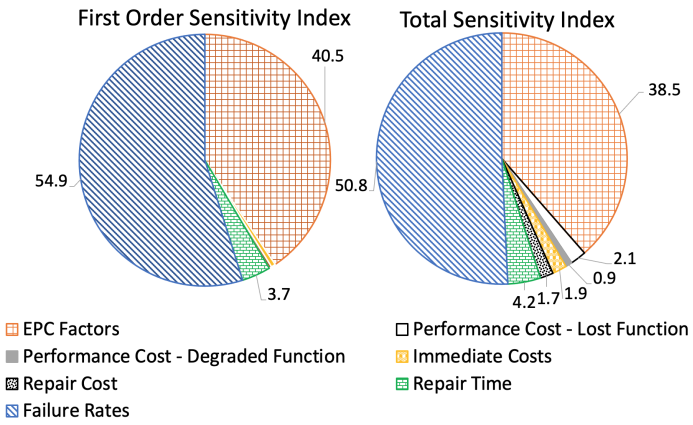


Figure 4.9: Average sensitivity indexes for variable groups

order and total sensitivity indexes (66% and 61%). Among the proportion of effects of EPC factor variables, variables relating to the action Turn for inlet and outlet valves had the highest first order and total sensitivity indexes. Because of the high uncertainty in these variables, designers may focus on designing the system to both improve the EPC factors and lower the sensitivity in the cost model (thus making the cost assessment more accurate and reducing project risk). As shown, sensitivity analysis helps pinpoint variables that require attention when considering uncertainties. When used in design, the HEFFR analysis and corresponding sensitivity analysis can be repeated iteratively as the design changes to ensure risk-related design goals are fulfilled.

4.5.3 Discussion

The example presented above shows how using a probability and cost model in the HEFFR framework can extend its capabilities to quantify the severity of component failures and human errors acting alone and in combination, compare fault scenarios, and identify the worst-case scenarios in terms of overall impact. The results show that the generated data can be assessed further to examine the impact of faulty behavior modes, identify action combinations with the greatest potential for improvement, and pinpoint human actions that need further refinement. In the case of the coolant tank design problem, we were able to identify the high-cost high probability failures requiring individual attention, low-cost

high probability failures, and scenarios too rare to require further assessment. The faulty behavior modes relating to the inlet pipe and tank had a higher expected cost among non-human induced faulty behaviors. The expected costs of human induced faulty behaviors were also high. We also found that the likelihood of human induced faulty behavior modes can be reduced significantly by only assessing a fraction of the action classification combinations. Among the action classification combinations with the highest potential for faulty behavior likelihood reduction, cognitive errors relating to actions Detect, Reach, and Turn were prevalent, though a few combinations also included non-cognitive errors relating to the action Turn.

The approach presented in this research is limited by the uncertainties present in the expected cost and likelihood calculations, including those in parameter estimates and modeled behaviors [196, 197]. This is an important limitation, because while some decisions may not need completely accurate inputs [198], design failures are theorized to result from designer biases [199]. A sensitivity analysis allows designers to account for some of these uncertainties (specially, uncertainty relating to parameter input variables) by pin-pointing variables that can have the highest effect on the system when uncertainties are present. This enables designers to account for these uncertainties and make better-informed decisions. The sensitivity analysis performed here showed that variables that are directly influenced by design decisions (failure rates of components and EPC-related variables) had a high impact on the expected cost when compared to cost-related variables, showing the importance of the designer's role when it comes to risk mitigation. The fact that both human error- and component failure-related variables having high total sensitivity indexes (or the effect of individual variables and the effect of all interactions) shows the importance of assessing both human errors and component failure in combination rather than in isolation. The results also show that performing a sensitivity analysis will allow designers to pinpoint specific variables or areas of design that need to be focused on to improve overall risk effectively.

The above results show how the introduction of the likelihood of scenarios and expected cost to the HEFFR framework can aid designers to evaluate fault scenarios and take risk mitigation action. Without these metrics, there is no way to distinguish between fault scenarios in the output

of the HEFFR framework, which creates the necessity to consider all scenarios equally. Since these simulations produce millions of potential fault scenarios that cause critical function failure, considering all fault scenarios to be equal is not feasible. Using these metrics in the HEFFR framework, fault scenarios can be prioritized based on their severity, enabling designers to prioritize the most important scenarios when designing mitigating features. In summary, this approach helps designers understand the impacts of component failure and human errors acting alone or in tandem in the early design phase to make risk-informed decisions. This is important because in traditional risk assessment methods, such vulnerabilities come to light later–when design changes are costly and time-consuming–forcing designers to find workarounds and retrofit changes (to meet deadlines and cost targets) rather than proactively guarding against such vulnerabilities by design. The early design application of the proposed approach reduces the chances of making such costly and time-consuming design changes.

When failure occurs, often, the impact on the operators and the surroundings is much higher compared to the impact of lost performance. Hence, it is important to understand how failures affect the environment and the human to be able to minimize risk appropriately. The proposed approach includes the immediate costs in the cost calculation model, enabling the quantification of the detrimental effects of failures on the environment and the safety of the human. With the proposed approach, we try to generate as much data as possible. With the advances in the field of data science, we believe that designers should be able to leverage as much data as possible to extract the information they need to solve a design problem. For example, the case study presented in this research tries to identify worst-case fault scenarios and impacts of faulty behavior modes and human actions. The data analysis presented in the results section was tailored to address these questions. Others may want to use this framework to compare design alternatives; for this, the expected cost of each can be used to trade design risk with other performance attributes (e.g., efficiency). For example, if one wishes to consider automating a process, system designs with and without human-component interactions can be assessed on the basis of expected cost. Similarly, if concept refinement and component selection are desired, the component behavior mode costs and probability can be assessed to identify points of potential improvement.

On the human front, designers might want to identify safe operating procedures, training requirements, or safety protocols, which all can be identified by analyzing the human actions and human induced behaviors of components. Given the amount of data present, the potential ways to analyze the data are not limited to what is listed here–the data can be synthesized to address design problems as designers see fit. All the benefits listed above, especially the data assessment requirements, encourage designers to think more deeply about the system under development early in the design process, which can result in well-thought-out designs. As a result, the potential for identifying vulnerabilities relating to human interactions and components later in the design stages or even after the system is in use is minimized.

One limitation of the model used in this work is that it assumes independence in the probability calculations, which may be an underestimate. The fact that a majority of scenarios as shown in Fig. 4.5 were given very low probabilities was a result of the underlying probability model form, which is subject to mathematical model uncertainty [200]. Thus, while using expected cost is shown here to help identify the highest-priority scenarios, valuing the set of scenarios remains a challenge because of the effect of epistemic model and parameter uncertainties [196]. However, in the early design stages, establishing dependencies to any reasonable accuracy is difficult, especially for human interactions. Hence, we recommend the use of the proposed metrics only to compare between scenarios and alternative designs rather than using them to quantify exact likelihood and cost. It should be noted that many of the later design stage probabilistic risk and safety assessment methods such as ETA, FTA, THERP, and SPAR-H incorporate dependencies in the probability calculations. Hence, the application of these methods later on in the design will allow engineers to understand the likelihood and cost of failure more accurately. Nevertheless, identifying the ideal underlying cost function and probability model to use in early design remains a challenge, and the use of different probability model assumptions should be explored in future work to determine the sensitivity of the value of these scenarios to model assumptions and identify the most appropriate model forms.

Also, the user defined proportion of effects of EPCs can be subjective. Previous work has attempted to remove the subjectivity surrounding this variable by replacing it with fuzzy linguistic

expressions [47]. A similar approach can be taken if no subjectivity is desired. In summary, with the introduction of the risk metrics, designers can use the HEFFR framework to identify worst-case fault scenarios, perform trade-off studies, establish operating procedures and training, identify points of potential human product interaction, and many more early in design. However, the probabilities calculated may be subjective or be an underestimate due to assumptions made. As a result, we advise using this framework to complement, not replace, traditional probabilistic risk assessment methods.

4.6 Conclusion

This chapter introduced an automated scenario generation approach and a risk quantification model to the HEFFR framework. The goal of this work is to allow designers to generate a wide range of potential fault scenarios involving human and components, and identify and prioritize worst-case fault scenarios. The majority of the scenarios generated had both human- and component-related vulnerabilities. Similarly, the application of the risk model showed that the interaction effects of component and human vulnerabilities had the highest sensitivity indexes. These result prove the importance of assessing the combined effects of human errors and component failures. With automated scenario generation and risk quantification, designers can use the HEFFR framework to identify worst-case fault scenarios, prioritize fault scenarios, quantify the impact of human errors and component failure, and pinpoint areas (both component and human interaction related) where improvements can yield the greatest risk mitigation. Additionally, the framework can be used for risk-based trade-off studies, to establish operating procedures and training, and to come up with safety protocols.

For the automated scenario generation, the study presented in this chapter only checks if one critical function is failing in a simple problem. Hence, the reported execution time and the failure scenarios do not particularly shed light on the overall performance of the algorithm. As future work, this approach should be applied to a more complex system with multiple executions and the results

should be used to improve performance metrics such as execution time and the number of scenarios executed. Another area of future work should explore results from multiple executions to look into ways to streamline the transition rules so that the number of scenarios executed is minimized while optimizing scenario coverage to include a majority of component failures and human errors. Streamlining the transition rules will help with improving the overall performance of the automated scenario generation approach introduced in this chapter. One limitation of the risk quantification model is that it does not consider the uncertainties when calculating expected cost and likelihood. While performing a sensitivity analysis (as presented here) can help designers to account for some of the uncertainties, it does not give designers a full picture of the effects of the uncertainties present within the model. Future work should study how to understand the effect of uncertainties on the model to enable designers to best account for it in hazard modeling and risk-based decision making.

Chapter 5: Evaluating the Performance of the Framework

This Chapter addresses research objective 3 by evaluating the performance of the Human Error and Functional Failure Reasoning (HEFFR) framework introduced in Chapters 3 and 4. The evaluation studies the framework's applicability to complex engineered systems and the validity in terms of its ability to predict and prioritize failures realistically. To study the applicability of the HEFFR framework to complex engineered systems, a modular risk assessment approach is introduced as means of managing complexity. Then, the modular risk assessment approach is validated for consistency to make sure that it can produce similar design insights to integral assessments and be consistent regardless of how the system is partitioned. This research has been accepted for publication in the Proceedings of the 2021 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference and was coauthored by Lukman Irshad, H. Onan Demirel, and Irem Y. Tumer [201]. This research will also be submitted for publication in ASME Journal of Computers and Information Science in Engineering. The validation study aims to validate the HEFFR framework in two fronts. First, the pros and cons of the HEFFR framework is explored in terms of its comparability with existing risk assessment methods. Next, the results from the application of the HEFFR framework to a train locomotive design study is compared with past train accidents to study if the HEFFR framework is capable of predicting those real-world failures and assigning appropriate severities. This research will be submitted for publication in the Journal of Mechanical Design and will be coauthored by Lukman Irshad, H. Onan Demirel, and Irem Y. Tumer.

In this chapter, we also demonstrate an application of the HEFFR framework to perform risk-based ergonomic assessments during early design stages using digital human modeling. The risk-based digital human modeling approach brings together the Human Error and Functional Failure Reasoning framework and digital human modeling platforms to prioritize ergonomic vulnerabilities and perform

ergonomic assessments. The approach is demonstrated using a train locomotive design case study and the digital human modeling platform Siemens Jack. Salman Ahmed, H. Onan Demirel, and Irem Y. Tumer contributed to this research.

5.1 Motivation

As with any new fault modeling framework, the HEFFR framework needs to be validated to understand if it can achieve its intended functions. The HEFFR framework was developed to enable designers to predict and prioritize failures (especially failures that result from component and human vulnerabilities interacting) early in the design stages. While the hold-up tank example presented in Chapters 3 and 4 does a good job in terms of demonstrating the ability of the HEFFR framework to generate fault scenarios involving components and humans, model the combined effects of component failures and human errors, and prioritize worst-case faults, it does not prove the scalability and the accuracy of the HEFFR framework. Since the automated scenario generation uses an exhaustive search, HEFFR simulations run the risk of becoming computationally expensive when applied to more complex problems. Also, since the HEFFR framework assigns probabilities and costs for failures based on the minimal information available during early design stages, the results are subject to high uncertainties. Hence, it is important to understand if the scenario generation and prioritization are accurate given the uncertainties present. This chapter aims to understand these paradigms by studying the application of the HEFFR framework to a complex engineered system and by comparing the results from the HEFFR framework with historic failures for accuracy. Then, the HEFFR framework is compared against existing risk assessment methods to establish its capabilities and limitation. Finally, an application of the HEFFR framework to perform risk-based ergonomic assessments during early design stages is demonstrated to show an example of how designers may use the HEFFR framework for design.

This research explores the applicability of the HEFFR framework to more complex problems by applying it to a train locomotive design case study. We explore a modular analysis approach

to manage complexity. First, the HEFFR framework is adapted to allow modular risk assessments. Then, we study the validity of the adaptations by exploring the following questions. Is it better to take an integral approach or a modular approach to analyze the design? Can the HEFFR framework give consistent results regardless of the mode of assessment? What are some risk-related insights designers can gain about the locomotive design? The results show that the modular assessments can significantly reduce the number of function evaluations and, as a result, computational costs while producing consistent results. The train locomotive study shows that the modular assessments can produce similar risk insights to integral assessments, and such insights need to be viewed through a modular context.

To validate the HEFFR framework, this research compares real-world train crashes that involved injuries or fatalities with the results from the HEFFR assessment of a train locomotive design study. Specifically, we explore the following questions. Is the framework able to automatically generate the scenarios that led to real-world accidents? If it was generated, what is the assigned severity when compared to the rest of the scenarios generated? Is the assigned severity appropriate when compared to the real-world outcome? If the designers of the system were to use this framework during the design of the system, would they have been able to catch and mitigate the potential accident? In addition to validating against real-world accident data, we also compare the HEFFR framework with existing risk assessment methods to understand its capabilities and limitations and the HEFFR framework's usage.

To demonstrate an example of how the HEFFR framework can be used to inform design, it is coupled with digital human modeling to perform risk-informed ergonomic assessments. Traditionally, task analyses are used to identify ergonomic vulnerabilities and identify the needed ergonomic studies. Usually, experts perform a task analysis and decide the ergonomic studies that need to be performed. Complex engineered systems may have a large number of potential human-machine interactions, meaning a large number of ergonomic studies may need to be performed. Additionally, designers may have to make trade-offs between different design features that may enhance some ergonomic features while negatively affecting others. During the early design stages, when minimal product details

are available, it might be challenging for experts to identify and prioritize the types of ergonomic assessment needed for a complex engineered system design because of the high uncertainties present. Additionally, when trade-offs are needed between ergonomic features, it may be hard for experts to make an informed decision when there are minimal data and a large number of ergonomic needs. This research demonstrates an application of the HEFFR framework to overcome these limitations. This HEFFR framework will be used to define and prioritize the needed ergonomics assessments based on the potential risk of faulty human-machine interactions. If the prioritized human ergonomic vulnerabilities are non-cognitive, digital human modeling is used to visualize the human product interactions and perform ergonomic assessments. The proposed application of the HEFFR framework is demonstrated using a locomotive design case study and Siemens Jack (a DHM tool), and the results are explored to understand the design insights that can be learned.

5.2 Background

This section forms the basis for using modular HEFFR assessments as a means of tackling complexity in complex engineered systems design.

5.2.1 Modularity in Engineering Design

We have proposed modular risk assessments as a means of managing system complexity in this research. A module is defined as a unit with strong internal connections and relatively weak external connections [202–204]. In other words, elements within a module have strong connections among them while having a weak interface with other modules [202, 205]. Modules work together to achieve the functions of a system while maintaining a certain degree of independence [202]. Modularity in design can help designers manage complexity by breaking down tasks into more manageable chunks [204]. Modular designs can reduce cost and design time by allowing parallel work [204, 206]. When modules are properly defined, modularity can promote innovation [207]. Modularity also

enables the mass customization and reuse of components [205]. It also allows faulty components to be replaced rather than replacing the system [206]. While these advantages show the promise of modularity in design, there are some disadvantages. Over modularization can increase testing and system integration time while hampering innovation [207]. Also, modularity can lead to trade-offs with performance [208] and robustness [206, 209, 210]. In summary, to fully reap the benefits of modularity, designers should avoid over modularizing.

In this research, we do not pursue modular or integral designs. Instead, explore modular analysis as a means of managing the complexity of performing risk assessments of complex engineered systems using the HEFFR framework. The use of modular analysis is agnostic to whether the designers are pursuing modular or integral designs. For modular risk assessments to be accurate, the overall system model needs to be partitioned into modules in a way that each module satisfies the definition of modules. There are several methods to partition modules. For instance, the Design Structure Matrix (DSM) method [211] is a matrix-based approach that uses a clustering algorithm to generate candidate modules. The Modular Function Deployment (MFD) method [212] uses 12 modularity drivers (e.g., technology evolution, planned product changes, etc.) and functionality to form modules. Other methods use network theory [213], fuzzy logic [214], genetic algorithm [215], and atomic theory-based clustering algorithm [216] to generate modules.

We recommend using the function structure heuristic method [217] to form modules when performing risk assessments using the HEFFR framework because the functional decomposition in this method is the same as the functional model in the HEFFR framework. This method uses three heuristics (dominant flow, branching flow, and transition modules) to devise modules from the functional decomposition of a product. When a flow enters the system or initiates, all subfunctions it passes through until it converts to a different flow or exits the system constitute a module according to the dominant flow heuristic. The branching flow heuristic defines all parallel function groups as modules. When a flow is converted and transmitted, the subfunctions involved in the process are defined as a module according to the transition modules heuristic. While this research encourages the use of the function structure heuristic method to generate modules, in circumstances where its use is

not appropriate, any other method that suits the situation can be employed to derive modules. For example, if a large system with thousands of subfunctions is to be analyzed, applying the heuristics manually can become cumbersome. In such a case, designers may use a method with automated module finding capabilities (e.g., DSM).

5.3 Applicability to Complex Engineered Systems

This study aims to present the applicability of the HEFFR framework to complex engineered systems. To achieve this, modular risk assessments are proposed to manage complexity and minimize computational expense. First, the approach to using the HEFFR framework for modular risk assessments is defined. Then, the proposed approach is validated by exploring whether the HEFFR framework can generate consistent results regardless of the mode of assessment (modular or integral) and the manner of partitioning. Further studies explore if the modular risk assessment approach can generate similar design insights to integral assessments. Note that details on the HEFFR framework, automated scenario generation, and risk quantification are not discussed in this section because they are discussed in detail in previous chapters. Instead, this section will focus on detailing how to adapt the HEFFR framework for modular risk assessments.

5.3.1 Methodology

5.3.1.1 Modular Risk Assessments Using the HEFFR Framework

The first step to performing modular risk assessments using the HEFFR framework is to partition the system into modules. We propose using the function structure heuristic method (discussed in section 5.2) to partition the functional model into modules. The overarching function is then defined for each module based on the specific function they perform. For example, for a module that converts and transports an energy flow, the overarching function is to convert energy. As shown in Fig. 5.1, a

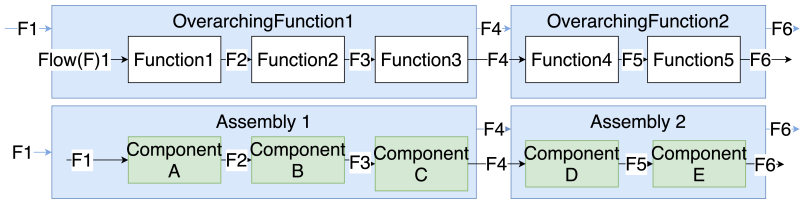


Figure 5.1: Generic module representation from the HEFFR system model

functional model of modules is created using the overarching functions. Next, the components in the configuration flow graph are clustered based on the function modules. For instance, if components A,B, and C in the configuration flow graph fulfill all functions in a function module, they are clustered into one module (refer Fig. 5.1). The component clusters act as individual configuration flow graphs for each module. As shown in Fig. 5.1, these component clusters are then combined into assemblies to create the configuration flow graph of modules.

The next step is to define the behavior model and the functional flow logic of the system. The behavior model and the functional flow logic are defined at both component and module levels. Each module is treated as a separate system when deriving the behavior model and function failure logic at the component level. Similar to component behavior models, the input-output relationships of the flows that pass through modules define the behavior model of component modules. For example, for an engine assembly, if the behavior mode is nominal, the mechanical energy output is proportional to the fuel input flow rate (e.g., if the input flow rate is nominal mechanical energy output is nominal). The functional failure logic classifies each overarching function as "operating," "degraded," and "lost" based on the input-output flow states of each overarching function. For modules with human interactions, the actions sequence graphs and action classifications are generated only at the component level. Human interactions with components may vary from component to component. Besides, humans may interact with multiple components in a module. Hence, if a module has multiple components with human interactions, they have to be modeled at the component level to be fully represented. Also, the ways humans interact with the system do not change if the components are clustered into a module. They still need to interact through the components. Hence, it is not necessary to derive action sequence graphs and action classifications for modules.

Fault Simulation and Results. We propose performing the fault simulation at both component and modular levels to quantify the severity of failures. In the HEFFR framework, an event is a combination of component behavior modes that occur at a time step. A scenario is a combination of events. At the component level, the HEFFR simulation is performed for each module separately. For each module, a HEFFR assessment of all possible combinations of component behavior modes (events) is performed to identify the events that can contribute to each output flow state of the module based on the inputs. The inputs may include the input flow of the module, control signals, or the behavior modes of components with dependencies with prior events. Some events may have dependencies with previous events. For example, if the behavior mode of a valve is "stuck" in the current event, its position in the current event is equal to its position in the previous event–if the valve is closed in the last event, it stays closed in the current event. Hence, we use behaviors from the last event as inputs for components that have such dependencies. At this level, the outcomes of individual events are evaluated with no time dependence.

The HEFFR simulation defines the control signals based on the system state. Since the component level assessments are performed for individual modules separately, and control signals that act as inputs to modules may not always be produced in the same module, it is not possible to define control signal states when the HEFFR assessment is performed at the component level. Hence, when the component level assessments are performed for modules, the events are evaluated separately for all combinations of the inputs. For example, if a module takes a control signal (e.g., True, False), the behavior state of a component in the last time step (e.g., State 1, State 2, and State 3), and an energy flow (None, Low, Nominal, High) as inputs, all events are analyzed for all combinations of the inputs (i.e., [True, State1, None], [True, State1, Low],....., [False, State1, High]). The total number of outcomes will be equal to the Number of Events $\times$ Number of Input Combinations.

The probabilities and expected cost of failures of module outputs are calculated using the probability and cost calculation model of the HEFFR framework. The event probabilities are calculated using failure rates, failure mode distributions, and human error probabilities (for details, refer section 4.5). Each module output can result from multiple individual events. Hence, to compute the likelihood of

a module output occurring for a given input, the probability of at least one event resulting in that specific outcome given the input is calculated (i.e., if events A,B, and C result in outcome1 if inputs C and D are present, P(outcome1, given C & D) = P(AUBUC)). The expected cost of an event is calculated using immediate cost, lost performance cost, recovery time, and repair cost (for details, refer section 4.5). The expected cost of a module output for a given model input is then calculated by taking the average of the expected costs of all events that result in that specific output for the given input.

With the probabilities and expected costs of module outputs, an event-time-based HEFFR simulation with automated scenario generation is performed at the modular level by treating modules as components and overarching functions as the functional model. For each module behavior mode, the probability and the expected cost of the current behavior mode is equal to the probability and the expected cost of the output, given the current inputs calculated at the component level simulation. At this level, the likelihood of an event occurring is calculated using the module behavior mode probabilities. Then, the scenario probabilities are derived based on the event probabilities. The expected cost of a scenario is calculated by considering component and module level failures. The contribution of components is calculated based on the event in the last time step, where the expected costs of each module output for the given inputs are summed. Modules may have functions different than the functions of their components. For example, a brake assembly slows down or stops the vehicle, which is a function that the components of the assembly do not fulfill individually. Hence, these modules can have additional immediate costs and performance costs. On the other hand, repair cost only exist at the component level because repairs are only performed on components and not on modules. To consider these module level failure costs, the expected cost of the module level function failures is calculated (using scenario probability and immediate and performance cost of lost module level functions). Then the expected cost of failure of a scenario is calculated by adding the component and module level expected costs. The module level expected cost and probabilities are calculated using the probability and cost model of the HEFFR framework using the underlying principles described above.

5.3.1.2 Validation Study

For the HEFFR assessments to be reliable, the results should be consistent regardless of the analyses being modular or integral. Also, the way the modules are divided (as long as they are well-defined and consistent) should not affect the results. We study the consistency of the results by performing two types of analyses: integral versus modular and modular versus modular. We do not expect the absolute values of the risk matrices to match between modes of assessment because of how the analysis is performed between different modes. Instead, we study if the results can shed consistent insights. For example, the components with the highest cost should be the same regardless of the mode of assessment, even though the actual costs may vary.

For the integral versus modular analysis, we assess a system integrally and modularly and compare the risk insights for consistency. Regardless of the mode of assessment, engineers should be able to gain insight into the most vulnerable components and behavior modes. In an integral assessment, this can be understood by calculating the cumulative expected cost of each behavior mode (if one desires to learn the most vulnerable component behavior modes) or each component failure (if component level information is sufficient). When performing a modular level analysis, such cumulative costs will only shed light into the vulnerabilities of module behavior modes and module failures. To understand the component level vulnerabilities, the contribution of each component behavior mode to the module behavior costs need to be calculated. One can achieve this by first calculating the average expected cost of each component behavior mode for every behavior mode. Then, these average expected costs of component behaviors of each module behavior modes can be summed according to Eq. 5.1 to calculate the average contribution of each component behavior mode to the system level failure costs. In Eq. 5.1, AC_b is average contribution of a component behavior mode to the system level failure costs, MB is module behavior mode, E is events, C_b is expected cost of the event with the behavior mode, and N is number of events.

$$AC_b = \sum_{i \in MB} \left[\frac{\sum_{j \in E} C_{b,i}}{N_j} \right] \tag{5.1}$$

When an integral assessment is performed, some behavior modes of components are more likely to appear in events than others. For example, if a pipe is leaking, the system may continue to operate in a degraded state, with the leak continuing in the following time steps. On the other hand, if the pipe is ruptured, the system will fail, and the simulation will move on to the next scenario. Hence, the behavior mode leak will be present in more scenarios than the behavior mode rupture. In a modular assessment, since the time-based simulation is only performed at a modular level, the above phenomenon is not captured at the component level. Also, the effects of some of the failure costs (that result from module function failures) are only captured at the modular level, while in integral assessments, they are captured at the component level. Because of these differences in the cost model, component behavior mode costs calculated from the integral assessment and modular assessment cannot be directly compared. Instead, we compare the expected failure cost of a component failing because behavior mode dependencies between components are represented in both integral and modular assessments. Hence, in both assessments, we calculate the expected cost of a component failing by taking the average of all faulty behavior mode expected costs of a component–cumulative expected cost of behavior modes for the integral assessment and average contribution of a component behavior mode to the system level failure costs for the modular assessment.

Since the modular assessments capture some of the failure costs at the module level, when comparing component failure costs, we compare them independent of other module components. For example, if a module has components A, B, and C, and another module has components D, E, and F, we compare the failure costs of A, B, and C and D, E, and F separately with the results from the integral assessment. We rank A, B, and C from the modular and integral assessments based on their failure cost, and the rank of D, E, and F are compared independently. For the modular versus modular assessment, we repeat the above study but with models with different module partitioning. We also compare the number of function evaluations to understand the computational costs of different modes of assessments. Finally, we explore the results of the modular risk assessment of the system to understand if the same insights gained from an integral assessment can be extracted through this mode of assessment.

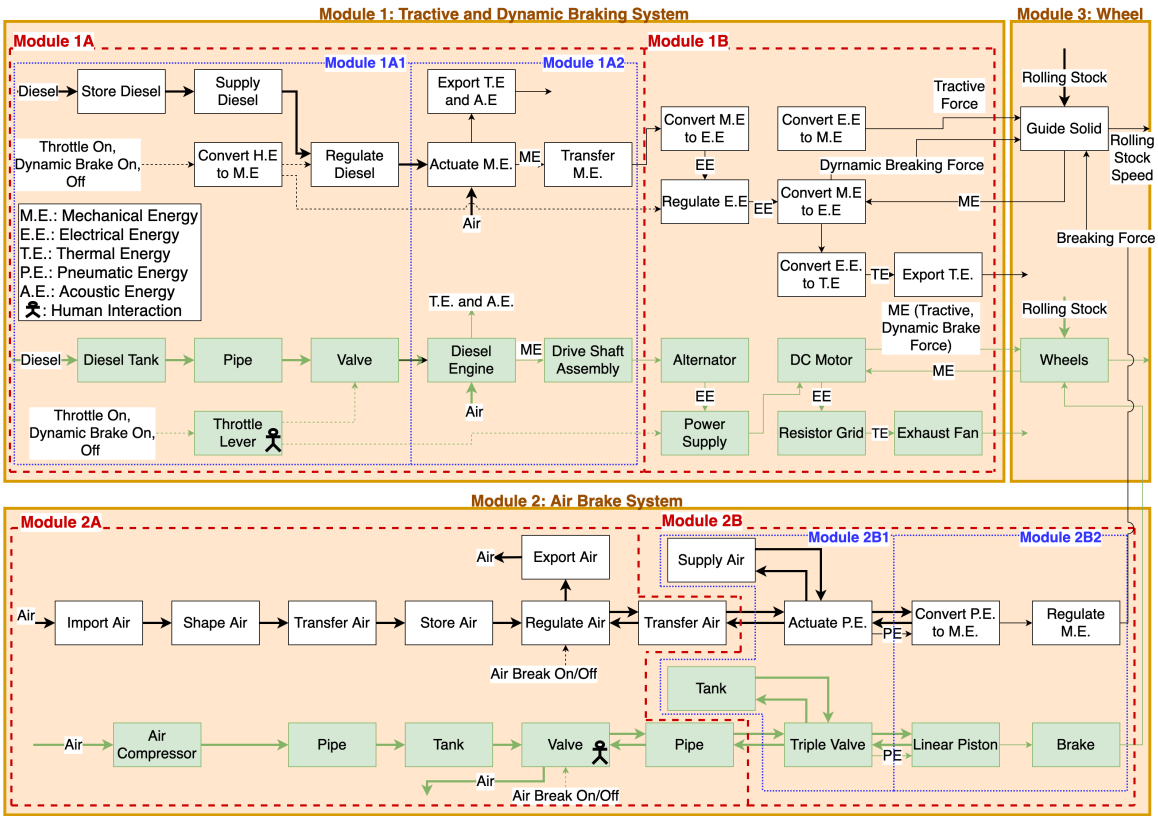


Figure 5.2: The functional model (White) and configuration flow graph (Green) of the train locomotive subsystems with module partitioning

5.3.2 Case Study: Diesel-Electric Locomotive

This research uses a train locomotive case study to demonstrate the applicability and validity of the HEFFR framework when assessing the risk of complex engineered systems. Specifically, we have conceptualized the air brake system and the throttle and dynamic brake system of a diesel-electric locomotive. The air brake can be operated by the train driver using a valve. The operator can interact with the tractive motor through a lever in the cab. Pushing the lever to the forward position activates the throttle (tractive force) while pushing it backward activates the dynamic brake. Leaving the lever at the center position does not generate any tractive force or dynamic braking force, which is equivalent to idling in "neutral" gear in automobiles. The control signals to apply the air brake,

dynamic brake, or throttle are triggered by the current locomotive speed and external stimuli. The external stimuli are fed into the system simulation requiring the train to stop, slowdown, or move. Each event is simulated for all three stimuli. Note that the locomotive operation has been simplified to replicate an early design stage conceptual design. Since the HEFFR framework is an early design stage risk assessment tool, without such simplifications, we cannot fully explore its ability to predict failures in complex engineered systems with the minimal information available during the early design stages.

The functional model (white), configuration flow graph (green), and the module partitioning of the system are shown in Fig. 5.2. In total, the system has 28 functions and 21 components, out of which two (throttle lever from the tractive and dynamic brake system and valve from air brake system) interact with the human. The actions and action classifications of the components that interact with the human are listed in Table 5.1. The components have 75 behavior modes in total, out of which ten are human induced. The system is partitioned into three main modules: Tractive and Dynamic Brake System, Wheel Assembly System, and Air Brake System. The Tractive and Dynamic Brake System is partitioned into two modules (Module 1A and Module1B, according to Fig. 5.2). Module 1A is further divided into two modules (Module 1A1 and Module 1A2). The integral versus modular study treats Module 1A as a whole system and Modules 1A1 and 1A2 as modules of that system. The Air Brake System is divided into two modules (Module 1A and Module 1B, according to Fig. 5.2). Module 1B is partitioned into two more submodules (Module 2B1 and Module 2B2). The modular versus modular analysis considers the Air Brake System with two modules (Modules 2A and 2B) and three modules (Modules 2A, 2B1, and 2B2) to perform the comparative study of different modularizations. Examples of functions, corresponding components, and their behavior modes at both component and modular levels are presented in Table 5.2. The human induced behaviors of components and assemblies are highlighted with **bold** text.

We used the three high-level subsystems (Modules 1, 2, and 3) as modules to assess the risk of the overall train locomotive design. We sourced the component failure rates and the failure distributions of behavior modes from Non-electronic Parts Reliability Data-95 (NPRD-95) [192] and

Table 5.1: Actions and action classifications from the action sequence graphs for the throttle lever and brake valve

Component (T: Throttle Lever and B: Brake Valve)	**Actions**	**Action Classifications**
T, B	See	Visible, Not Visible
T, B	Detect	Detected-Nominal, Detected-False, NotDetected-Nominal, NotDetected-False
T, B	Reach	Reached - Nominal, Reached - False, Cannot Reach, No Action
T, B	Grasp	Grasped, Cannot Grasp, No Action
T	Move	Move to Throttle On, Move to Dynamic Brake On, Move to Off, Cannot Move, No Action
B	Turn	Turn to Open, Turn to Close, Cannot Turn, No Action

Table 5.2: Selected functions, corresponding components, and behavior modes at component- and modular-level

Function	**Module (Yes/No)**	**Component/Assembly**	**Behavior Modes**
Convert H.E. to M.E.	No	Lever	**Nominal Off, Nominal Throttle On, Nominal Dynamic Brake On, Failed Off, Failed Throttle On, Failed Dynamic Brake On, Stuck Off, Stuck Throttle On, Stuck Dynamic Brake On**
Actuate M.E.	No	Diesel Engine	Nominal, Loss of Control, Failed
Convert M.E. to E.E.	No	Alternator	Nominal, Drift Low, Drift High, No Operation
Regulate Air, Export Air	No	Valve	**Nominal On, Nominal Off, Failed Open, Failed Close, Stuck Open, Stuck Close**
Regulate P.E.	No	Disc Brake	Nominal, Degraded Operation, Failed
Regulate Air	Yes (Module 2A)	Air Pressure Regulation Assembly	**Nominal, Failed, Low Pressure, High Pressure**
Actuate P.E.	Yes (Module 2B1)	Brake Actuator Assembly	Nominal Actuation, Nominal No Actuation, Insufficient Actuation, Failed Actuation, Failed No Actuation
Regulate M.E.	Yes (Module 2B2)	Brake Assembly	Nominal Braking, Insufficient Braking, No Braking

Failure Modes/Mechanisms Distributions-97 (FMD-97) [193], respectively. The expected time of operation with no maintenance is selected as 6,700 hours, assuming that the train travels 100,000 miles per year at an average speed of 45 mph for three years. The recovery times of the components were estimated considering the total down-time (e.g., diagnosis and repair time). The cost of lost functions is estimated based on the impact of the functions being in a "degraded" or "lost" state on the overall system performance. The immediate costs of lost functions are estimated assuming that the locomotive is for a passenger train (e.g., safety, fatalities and injuries, loss of service of the track, etc.). Four HEART [25] error producing conditions (or performance shaping factors) were identified to influence the human action probabilities. They are,

- EPC2: a shortage of time available for error detection and corrections,
- EPC8: a channel capacity overload, particularly one caused by simultaneous presentation of non-redundant information,
- EPC10: the need to transfer specific knowledge from task to task without loss, and

- EPC34: prolonged inactivity or highly repetitious cycling of low mental workload tasks.

The error producing conditions were chosen based on the design of the locomotive and the actions humans will perform to interact with the components.

Using this case study, we first study the validity of modular HEFFR assessment by comparing the results between integral and modular assessments. Then, we explore the consistency and computational expense of the modular approach for varying module divisions. We finally study results from the modular assessment of the whole train model to study how it can guide risk-based design decision-making. For this analysis, we have chosen the function Guide Solid as the critical function because its loss means that the train is not performing its intended function, or worse–a derailment or a crash. For assessments on subsystems, primary functions that directly affect the Guide Solid function of the train were chosen as the critical function.

5.3.3 Results

For both integral versus modular and modular versus modular assessments, we performed a time-based HEFFR simulation. The total number of time steps was set to 3, and the number of times an event is allowed to repeat in a scenario was set to 1. The system was modeled in a way that if a failure were to occur, it would occur within the above time step limits. As shown in Tables 5.3 and 5.4, for both integral versus modular and modular versus modular assessments, the module level ranking of components based on the expected cost of them failing were consistent across assessment modes. For example, in the integral versus modular analysis, the component engine had the highest expected cost, followed by the component shaft for Module 1A2 in the modular assessment. The ranking of these two components was the same in the integral assessment when they were grouped. In the modular versus modular assessment, the ranking of the modules based on their expected cost of failures was also consistent. Module 2A had the highest expected cost, followed by 2B (and the 2B subpartitions in the case of the assessment with 3 module partitionings). In the integral versus modular assessment, the number of function evaluations for the modular assessment reduced

Table 5.3: Component rankings based on expected cost of component failure: integral vs. modular assessment

			Component/Module Expected Failure Cost (USD)		
			Modular Assessment		Integal Assessment
Module	**Rank**	**Component**	**Module Level**	**Component Level**	**Component Level**
Module 1A2	1	Engine	964,619	2419	2,168,066
	2	Shaft		8.32	18,963
Module 1A1	1	Lever	416,459	92.32	429,598
	2	Tank		90.20	118,975
	3	Pipe		4.28	22,159
	4	Valve		3.02	3598
Number of Function Evaluations			2,210		273,599

Table 5.4: Component rankings based on expected cost of component failure: modular vs. modular assessment

	Component/Module Expected Cost of Failure (USD)							
	2 Module Partitioning				3 Module Partitioning			
Component	**Rank**	**Module Name**	**Module Level**	**Component Level**	**Rank**	**Module Name**	**Module Level**	**Component Level**
Valve	1	Module 2A	362,126	3,341	1	Module 2A	27,766,293	3,341
Compressor	2			1,208	2			1,208
Tank	3			590.7	3			590.7
Pipe1	4			15.7	4			15.7
Pipe2	5			2.06	5			2.06
Tank	1	Module 2B	233,243	2,281	1	Module 2B1	22,850,524	11,054
Triple Valve	2			371.5	2			4,173
Piston	3			119	1	Module 2B2	22,803,748	2,773
Brake Assembly	4			64.5	2			16.9
Number of Function Evaluations		1,070			26,389			

significantly (by 99.19%) when compared to the integral assessment. However, as the number of modules increased from two to three in the modular versus modular assessment, the number of function evaluations increased by 2,466%.

For the analysis of the locomotive design, the total number of time steps is set to 3, and the number of time steps an event is allowed to repeat is set to 2 to accommodate the changes in the train speed. The total simulation took 218,002 function evaluations. Because the number of function evaluations grows exponentially as the number of behavior modes and time steps increase, the same simulation would have taken hundreds of millions (around 1 billion) function evaluations if an integral assessment was performed. As shown in Fig. 5.3, among the three modules, the Wheel System had the highest average expected cost followed by the Air Brake System and Tractive Force and Dynamic

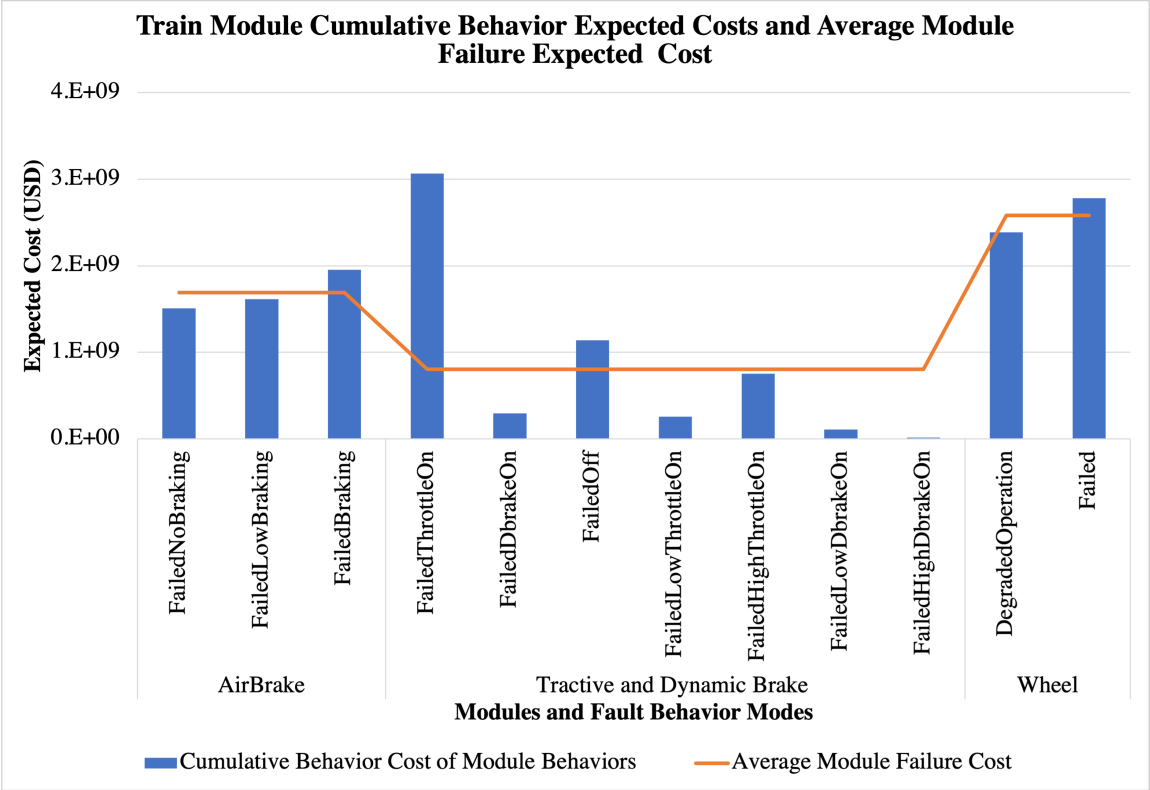


Figure 5.3: Train module behavior cumulative expected cost and average module failure expected cost

Brake System. The module behavior "failed throttle on" of the Tractive Force and Dynamic Brake System had the highest severity followed by "failed" and "degraded operation" of the Wheel Assembly System. To understand the contribution of components to these module failures, one may assess the average expected cost of failures of components in this module (by following the average contribution of a component behavior mode to the system level failure costs calculation model described in section 5.3.1, and taking the average of faulty behavior mode costs). However, this will not give insight into worst-case behavior modes like in integral assessments. Without that knowledge, one cannot gain insight into how to tackle specific behaviors, especially human induced, to mitigate risk.

For each module, one can identify components that contribute the highest to the systems failure cost by ranking them based on their average failure expected costs. Once the highest contributor

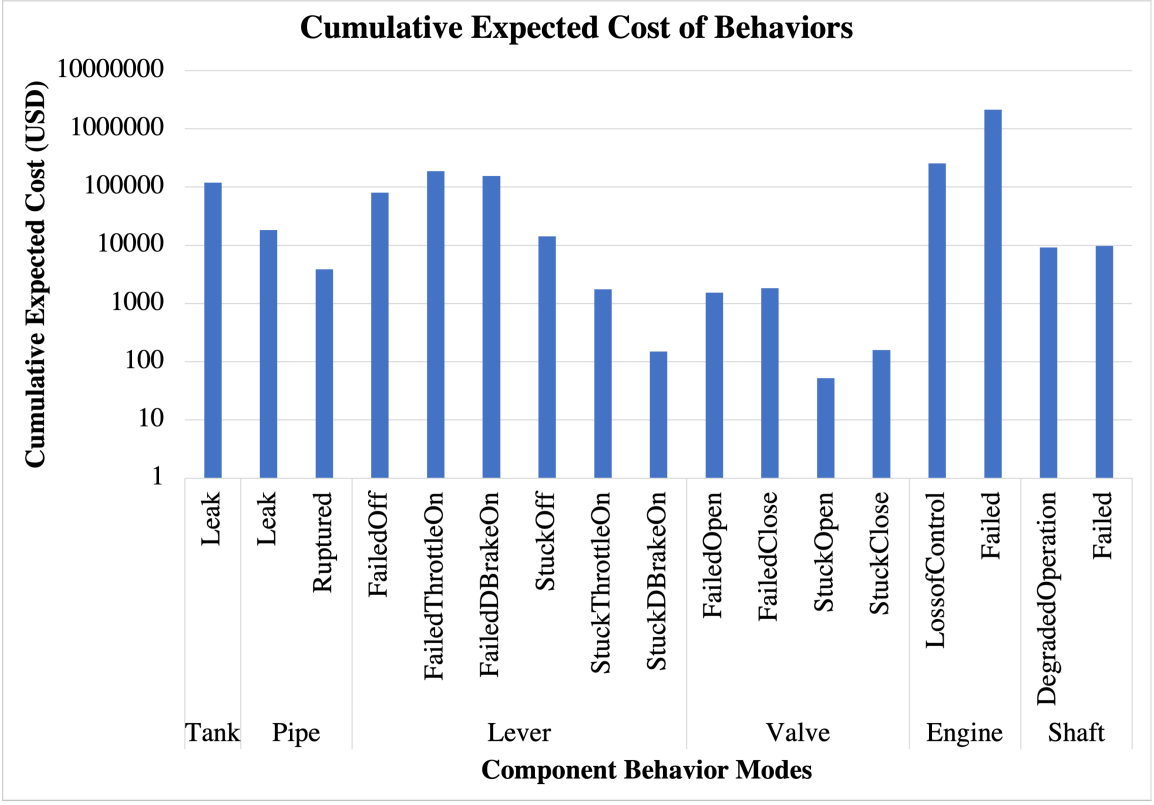


Figure 5.4: The cumulative expected cost of behaviors of components in module 1A: integral assessment

components are found, the sub-modules these components belong to can be found. Worst-case behavior modes can then be identified by performing integral HEFFR assessments on these sub-modules and taking the cumulative expected cost of each behavior mode. In the train locomotive design study, we chose the Tractive and Dynamic Brake System (the module with the behavior mode with highest cumulative expected cost) to perform this detailed assessment. In reality, designers may chose any module (e.g., modules with the highest average expected cost of failure) as they see fit based on their design needs. For the Tractive and Dynamic Brake System, the components diesel engine and lever were among the highest contributors to system-level expected cost. So, we performed an integral assessment of the module (Module 1A) these components belonged to by choosing the function Transfer ME as the critical function. The results from this analysis are shown in Fig. 5.4.

As shown in Fig. 5.4, the behavior modes “loss of control” and “failed” of the diesel engine have the highest cumulative expected costs, followed by behavior modes “failed throttle on,” “failed dynamic brake on,” and “failed off” of the component lever. The behaviors of the diesel engine are non-human induced. To minimize the expected cost of these behaviors, designers may choose components with lower failure rates, add redundancies, include advanced failure detection mechanisms, or suggest testing. The failure modes of the lever are human induced behaviors. To minimize the expected cost of these behavior modes, designers may assess the human action combinations like in Chapter 4 to identify worst-case human actions and action sequences. Then, based on the analysis, designers may suggest design changes to improve performance shaping factors, ergonomic assessments, training, operating procedures, or safety protocols. As design changes are made, these HEFFR assessments can be repeated until the risk-related design goals are fulfilled.

5.3.4 Discussion

The above results demonstrates that the HEFFR framework can be scaled to assess complex engineered systems using the modular analysis of the train locomotive design. The results show that the modular analysis approach can produce consistent results for the modules that were assessed regardless of the mode of assessment (integral versus modular and varying module partitionings). Also, the computational cost was significantly low for the modular assessment compared to the integral assessment. However, the computational cost increased (still lower than integral assessment) as the number of modules increased. Also, the results of the locomotive design study show that the module-based assessment can pinpoint similar design insights as integral assessments when combined with targeted module level integral assessments.

For both integral versus modular and modular versus modular analyses, the assessment was only performed on one subsystem each. As a result, we cannot conclude that the modular analysis will always produce consistent results unless more systems are tested. However, the results are sufficient to justify the use of the modular HEFFR assessment approach in the train locomotive design study

because the validation studies were performed on modules of this system. The advantage of lower computational cost faded as the number of modules increased. Hence, when performing modular assessments, careful attention needs to be given to the module partitioning. Past research has shown that the modularity of a system increases as the model granularity increases [218]. That means that in highly detailed system models, a large number of modules can be present. In such cases, to offset the risk of increasing computational costs, we recommend combining modules (e.g., combine Modules 1A1 and 1A2).

Having to combine the modular assessment with targeted module level integral assessments to gain detailed insight into the cost of behavior modes may seem tedious and computationally heavy, taking away the advantages of modular assessments. However, compared to having to perform an integral assessment on the whole system, performing such targeted assessments are significantly computationally cheaper. For example, in the case study presented in this research, including the targeted integral assessment of Module 1A, the total functional evaluations were close to 500,000 which is significantly lower than the estimated number of functional evaluations (around 1 billion) for the integral assessment. Hence, performing such targeted assessments is still advantageous compared to having to perform integral assessments. During the design of complex engineered systems, usually, different teams work on different parts of the design. In such cases, not all teams need to know all the details about the individual subsystems the other team is designing. The design teams can use the module-based analysis to get the bigger picture (in terms of system risk) of the design decisions of other teams, while the targeted assessments can help them perform more detailed studies on the subsystems they are designing.

Future studies may perform a more comprehensive validation study by expanding the integral versus modular and modular versus modular analyses to more systems to understand the validity of the modular HEFFR assessment approach to complex engineered systems in general. Also, the validation study only checked the consistency in the results for one metric (expected cost of component failures). While this metric can help designers determine important insights relating to component and human behaviors, it is not useful in assisting risk-based trade studies. To understand if the

modular analysis method can be used for risk-based trade studies, future research should include the expected cost of the overall system as an additional metric that will be tested for consistency in the above study expansion. For the purposes of this dissertation such an expanded study is not necessary because the next section will be validating the overall HEFFR framework including the modular risk assessments against real world accidents. As a result, any shortcoming in the modular risk assessment approach will be exposed in that study.

5.4 Validating the HEFFR Framework

This section aims to validate the Human Error and Functional Failure Reasoning (HEFFR) framework. First, we validate the ability of the HEFFR framework to generate and prioritize fault scenarios that represent real-world failures by comparing results from a HEFFR analysis of a train locomotive design with historic train crashes that involved fatalities and injuries. We specifically explore if the HEFFR framework is capable of generating the scenarios that led to past accidents. Since the HEFFR framework can generate a large number of potential fault scenarios, we further study if the severities assigned by the HEFFR framework would have helped designers identify and prioritize the scenarios that lead to train crashes. In addition, we explore if the HEFFR framework helps designers identify the behaviors (both human- and component-related) that contribute to failures so that appropriate mitigation strategies can be built into the system. In addition to validating the HEFFR framework against past train accidents, we compare it with existing risk assessment methods to understand how it can be used to complement risk-based design.

5.4.1 Methodology

The first step to validating the HEFFR framework is to extract past accident data to compare with the HEFFR results. This study will compare the results from the modular HEFFR assessment of the train design case study in section 5.3.2 with past train accident data. The train accident

Table 5.5: Train modules and module behaviors

Module	Behaviors
Air Brake System	Nominal No Braking, Nominal Braking, Failed No Braking, Failed Low Braking, Failed Braking
Throttle and Dynamic Braking System	Nominal Throttle On, Nominal Dynamic Brake On, Nominal Off, Failed Throttle On, Failed Throttle On - Low, Failed Throttle On - High, Failed Dynamic Brake On, Failed Dynamic Brake On - Low, Failed Dynamic Brake On - High, Failed Off
Wheel Assembly	Nominal, Degraded Operation, Failed

data were extracted from investigation reports from two databases: National Transportation Safety Board (NTSB) and European Railway Accident Information Links (ERAIL). Both of these databases (especially ERAIL) had reliable accident data only for accidents that happened after 2005. Hence, only final reports published after the year 2005 were considered. Considering all accidents since the year 2005 is not feasible because there is a large number of accidents, and analyzing each of them will not add value to the validation study. Hence, this research focuses on a subset of accidents–accidents with the highest severity. We defined the severity of accidents based on the number of fatalities and injuries. All accidents with more than five fatalities and ten injuries were chosen to be analyzed as part of the validation study. In total, 75 accidents were identified to fulfill these criteria. Among the 75 accidents, 48 were identified as accident scenarios that must be present in the HEFFR train design case study results. Accidents that were omitted included accidents in fully automated trains (since these have no human operators and modeling them will not add value), and accidents that were caused by elements outside of the train system boundary (e.g., traffic controller errors, track maintenance crew errors, road vehicle driver error in level crossing accidents, etc.).

The next step is to convert events that led up to the accidents into HEFFR fault scenarios. We do this by representing the events that led to the accidents through external stimuli ("move," "slow," and "stop") and module behaviors of the three modules (Air Brake System, Throttle and Dynamic Braking System, and Wheel Assembly System) that were modeled using the HEFFR framework. The module behaviors of each module are presented in Table 5.5. In addition to converting events into modules, the faulty component behaviors and human actions that contributed to the accidents were also identified. For example, the Santiago train derailment in Spain [219], which resulted in 80 fatalities and 144 injuries, was caused by overspeeding. The driver's attention was diverted with

Table 5.6: Example train accidents in the HEFFR fault scenario format

Timestep	Throttle and Dynamic Braking System Behavior	Air Brake System Behavior	Wheel Assembly Behavior	External Stimulus
	Train Derailment - Santiago, Spain (2013)			
0	Nominal Throttle On	Nominal No Braking	Nominal	Move
1	Failed Throttle On	Nominal No Braking	Nominal	Slow
	Train Derailment - Paulsboro, NJ (2012)			
0	Nominal Throttle On	Nominal No Braking	Nominal	Move
1	Nominal Dynamic Brake On	Nominal Braking	Nominal	Stop
2	Failed Throttle On	Failed No Braking	Nominal	Stop

repeated phone calls and caused him to brake too late when entering a curve. The late braking resulted in the train entering the curve at almost double the speed than the recommended speed limit. The investigation concluded that that the cause of the derailment was the lack of attention paid by the driver (purely human factors related). This portion of the track was not installed with an automated accident prevention system (which would have slowed down the train automatically to the required speed limit). However, it included a safety warning system for overspeeding. If the design decision to include the automated accident prevention system had been made, this train crash could have been prevented. In the HEFFR fault scenario form, when the accident occurred the external stimuli is "slow" (requiring the train to slow down), and the behavior modes of the Air Brake System and the Throttle and Dynamic Brake System are "nominal no braking" and "failed throttle on," respectively. The air brake is in a nominal state because at high speeds dynamic brakes are usually used to slow down the train and not the air brake. Hence, the driver's lack of use of air brake is modeled as a nominal state. At the component level, the throttle lever of the Throttle and Dynamic Brake System and the brake valve of the Air Brake System were in failed human induced behavior modes caused by the human action Detect being in a failed state.

Another accident, the freight train derailment in Paulsboro, New Jersey [220], had more than 28 injuries and was estimated to cost around 30.5 million U.S. Dollars. The train driver stopped at a red signal and waited for a movable bridge to extend and fall in place. The red signal usually turns green once the bridge is fully engaged. This time it did not turn green because the bridge was not engaged properly due to a malfunction. The operator visually inspected the bridge and confirmed that it was fully engaged and was given permission to bypass the red signal. The train

derailed because of the false detection by the operator. In the HEFFR fault scenario form, when the accident occurred, the behavior modes of the modules are the same as the Santiago accident, but the external stimulus is "stop." The HEFFR scenarios for both of the accidents above are presented in Table 5.6. Note that the train locomotive designs and the fault scenarios generated do not include any advanced features such as the safety warning system. The train design was kept simple to keep the design minimal, representing an early design stage concept. One of the main objectives of this validation study is to explore the abilities of the HEFFR framework during the conceptualization stages (where concepts may start from a basic form and developed over time) of design. Hence, it is important to validate with a basic concept when the uncertainties are at the highest.

With the converted train accident scenarios we aim to explore three questions.

- Q_1: Will the HEFFR train locomotive HEFFR assessment generate all of the accident scenarios.
- Q_2: If the accident scenarios were generated, did the HEFFR assessment assign severity that are high enough to enable designers to easily detect those accidents?
- Q_3: Are the most common module behaviors, component behaviors, and human actions that were involved in the accidents easily identifiable?

The goal of Q_1 is to validate if the HEFFR framework can generate a wide range of realistic fault scenarios. The answers to Q_1 are explored by searching through the HEFFR results to see if all of the accident scenarios were generated by HEFFR. The second question studies if the severity quantification in the HEFFR framework is realistic and if it does help designers identify worst-case fault scenarios accurately. To study Q_2, we rank the HEFFR fault scenarios based on severity and check if the accident scenarios are ranked in the top half. The third question validates if the HEFFR framework is capable of pinpointing the worst-case component behaviors and human actions. Q_3 is investigated by first ranking the cumulative expected cost of each module behavior and checking if the module behaviors that were involved in the accidents are ranked in the top 50^{th} percentile.

We calculate the cumulative expected cost of a module behavior by summing the expected cost of all scenarios with the module behavior present in the last time step. We only consider the last

time step for the cumulative cost of behavior calculation because the module behaviors of the train accidents are from the final moments of the accident and most reports do not detail the events that happened before. For each of the module behavior, we calculate the contribution of component behaviors and test if the component behaviors that contributed to the accidents are ranked among the top 50^{th} percentile. We calculate the contribution of component behaviors by taking the average event cost of each component behavior mode that can result in a particular module behavior. Similarly, we calculate the contribution of human actions to human induced behaviors and check if the human actions that contributed to the accident are easily identifiable in the HEFFR results. We calculate the contribution of human actions by counting the number of action combinations that can result in a human induced behavior with a specific faulty human action present. If the majority of the human action combinations that result in a specific faulty behavior have a specific faulty human action, we deduce that the designers will easily identify and prioritize that human action. The goal of these tests is to find out if the HEFFR framework will help designers identify the faulty module behaviors, component behaviors, and human actions that can have the worst outcomes. For Q_2 and Q_3, we assume that the designers will prioritize fault scenarios, module behaviors, component behaviors, and human actions that are in the top half when they are ranked in terms of their contribution to the severity of failures.

The final step in the validation study is to compare the HEFFR framework with existing risk assessment methods to understand its merits and shortcomings compared to existing risk assessment methods. The results of this study will give insight into the ideal usage of the HEFFR framework and when other methods are preferred over the HEFFR framework. To compare HEFFR with the existing risk assessment methods, we first identify the capabilities and limitations of the existing risk assessment methods discussed in Chapter 2. Then, we identify the capabilities and limitations of the HEFFR framework from the studies in Chapters 3, 4, and 5. Then, the results are compared to understand how the HEFFR framework compares against existing risk assessment methods. The results of the validation study are detailed in the following section.

Table 5.7: HEFFR accident scenario ranking based on severity and minimum and maximum expected costs of scenarios with the same end state as the accident scenario

Scenario Rank		Total Accidents	Number of Similar Scenarios Generated by HEFFR	Minimum Expected Cost Among Similar Scenarios (USD)	Maximum Expected Cost Among Similar Scenarios (USD)
1	3565	2	68	7340	25037
2	287	1	68	24621	166548
3	273	22	33	14682	717277
4	70	5	33	14682	717277
5	37	1	4	14815	1090563
6	22	1	10	7346	1266181
7	8	14	5	14864	2423381

5.4.2 Results

Seven unique HEFFR fault scenarios were created when all accident scenarios were converted to HEFFR form. As shown in Table 5.7, a majority of accidents (36) were represented by two HEFFR fault scenarios. While the scenarios that led to these accidents were similar at a higher level (e.g., failure to detect a signal and proceeding without stopping), they still had minor differences in details (e.g., late application of brake vs. no application of brake). Since HEFFR simulation is discrete, it is not able to capture such minor details. Since the outcome is not stopping at the expected moment in both late application and no application of brakes, HEFFR treats them similarly. This lack of fidelity is an expected trait of HEFFR because it is an early design stage risk assessment framework that models failures with the abstract data that is available.

Answering the first question formulated above, all of the accident scenarios were generated by the HEFFR framework. The accident scenarios also had a shorter time to failure 1 or 2 time steps, meaning that if designers were analyzing the fault scenarios with shorter time steps to failure, they would have identified and prioritized the accident scenarios. As shown in Table 5.7, the HEFFR framework also generated scenarios (with 1 or 2 time steps) that would have had the same result as an accident but with a different event sequence. For example, for accident scenario 1, the HEFFR framework generated 68 similar scenarios that had the same result with different event sequences in 2 or 3 time steps. The expected cost of these scenarios differs (even though they have the same outcome) because the probabilities of occurrence are different. All but one accident scenario had the highest expected cost among the similar accident scenarios. In the case where the accident scenario

cost was not the highest (Scenario 3), the similar accident scenario with the highest expected cost was captured as part of a different accident scenario (Scenario 4). Being able to analyze similar fault scenarios to worst-case fault scenarios will help designers study the different event sequences that may lead to the same system-level failures and make sure that preventive strategies are built into the design. When a large number of scenarios are generated, lower severity scenarios will only get minimal attention (often analyzed as a whole and not individually). However, having similar scenarios to worst-case scenarios will allow designers to identify and individually analyze lower severity fault scenarios that may have a similar result to worst-case fault scenarios.

As shown in Fig. 5.5, all of the accident scenarios were assigned serveries in the top 50^{th} percentile, answering the second question. The expected costs were ranked higher than the 97^{th} percentile of the 134,187 scenarios generated by HEFFR, meaning that 97 percent of the generated scenarios had expected costs that were lower than or equal to the accident scenarios. This is much better results than the expected 50^{th} percentile, and it means that the accidents scenarios will be among the very top priority (meaning higher chances of risk mitigation) if the scenarios are prioritized based on the severities. A majority of the accidents had the faulty module behaviors “failed no braking” for the Air Brake System and “failed throttle on” for the Throttle and Dynamic Braking System. Among the accidents that involved the Wheel Assembly System, the faulty module behavior “failed” was common. As seen in Fig. 5.6, the modules that are most common among the accidents scenarios had the highest cumulative expected costs (when considering their presence in the last step of the scenario) for each subsystem (highlighted in red). This means that the HEFFR framework can help designers identify and prioritize the faulty module behaviors that can have the worst outcomes.

The human-induced component behaviors “failed throttle on” of the throttle lever and “failed close” of the brake valve were the most common faulty component behaviors among the accident scenarios. The component behaviors “failed throttle on” of the throttle lever, “leak” of the diesel tank, and “stuck throttle on” of the throttle lever were the top three (among 27 faulty component behaviors) contributors to the module behavior “failed throttle on” of Throttle and Dynamic Braking System in the HEFFR framework results. The component behaviors “stuck close” of the triple valve,

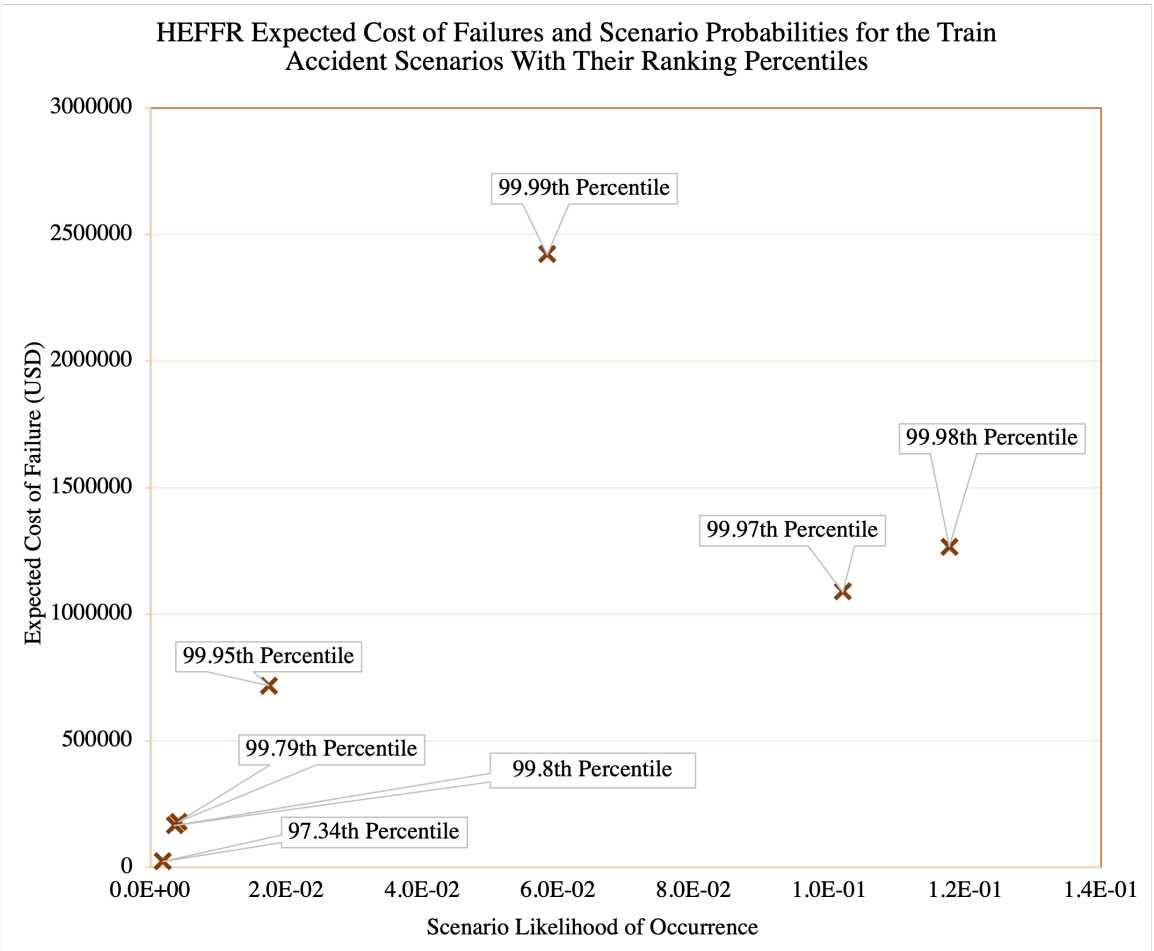


Figure 5.5: Expected cost of failures and likelihood of occurrence for the train accident scenarios with their ranking percentiles when compared to rest of the scenarios generated by the HEFFR framework

"degraded operation" of the brake assembly, and "failed close" of the brake valve were identified as the top three (among 18 faulty component behaviors) contributors to the module behavior "failed no braking" of the Air Brake System in the HEFFR framework results. Both common component behaviors in the accident scenarios were ranked higher than the top 50^{th} percentile (Ranked as the highest contributor in the case of "failed throttle on" and ranked third highest contributor in the case of "failed close"), meaning that the HEFFR framework would have helped designers pinpoint the faulty component behavior modes that had the worst outcomes.

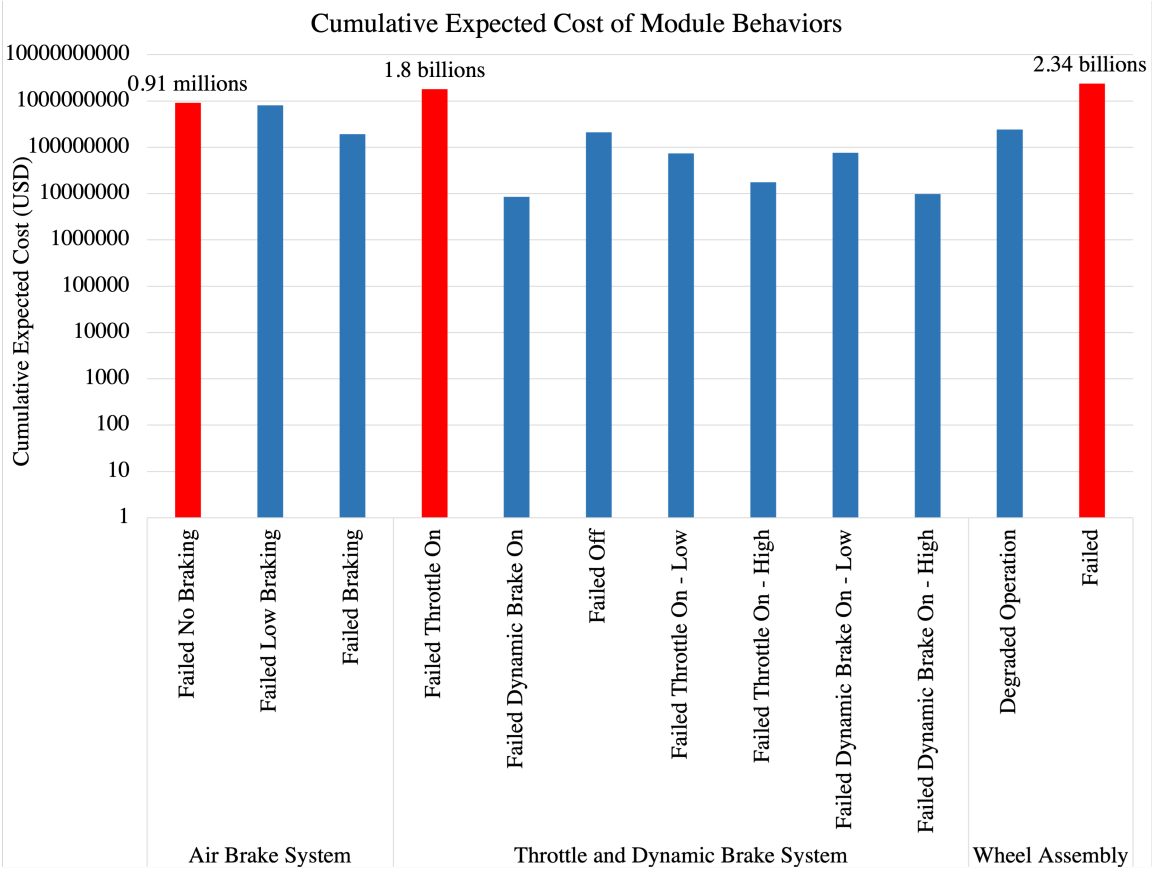


Figure 5.6: The cumulative expected cost of behaviors of modules with the fault behavior modes that were present in most train accidents highlighted in red

Action classifications “failed not detected” (signal present but not detected) and “failed detected” (signal is not present or a different signal is present and wrongfully detected) were among the most common human action failures that contributed to the faulty component behaviors that resulted in the accidents. The HEFFR framework assessment results showed that 36 human action combinations of the 48 that can result in the component behavior “failed close” for the brake valve had the faulty human action classification “failed not detected.” One hundred and twelve human action combinations of the 130 that can result in the component behavior “failed throttle on” for the throttle lever had either the “failed fetected” or “failed fot fetected” action classifications. The presence of the faulty

human actions that were common in the accident scenarios in a high percentage of action combination means that the HEFFR framework would have helped designers identify these faulty human actions with the worst system-level outcomes. In summary, these results show that the HEFFR framework would have helped designers pinpoint the faulty module behaviors, component behaviors, and human actions that would result in failures with the highest severity, answering the third question.

As shown in Table 5.8, the major advantage of the HEFFR framework when compared to existing methods is its ability to analyze the system level propagation of human errors and component failures acting in combination during early design stages. The HEFFR framework lacks fidelity when compared to the late design stage risk assessment methods. This is expected because the HEFFR framework models failures with the minimal data available during the early design stages. The risk matrices of the HEFFR framework are relative, making them useful only for comparison purposes, whereas the risk matrices in more detailed methods are more accurate and can be used to understand failure probabilities and severities better. HEFFR uses event-time when used with automated scenario generation and discrete-time when fault prediction is performed alone (as in Chapter 3) for dynamic simulations. The dynamic risk assessment methods, on the other hand, are capable of more detailed simulations with continuous-time. Theoretically, the HEFFR framework should be able to analyze hardware-software and software-human interaction related failures because it is an extension of the FFIP framework. However, this has to be validated to fully understand the capabilities of the HEFFR framework surrounding modeling hardware-software and software-human interaction-related failures. When compared with systemic risk assessment methods, the HEFFR framework falls short in terms of analyzing organizational factors related risks. In summary, while there are some limitations to HEFFR when compared to existing risk assessment methods, none of the existing methods are capable of analyzing the system-level effects of component failure and human error interactions during early design stages as the HEFFR framework does.

Table 5.8: Comparing the capabilities and limitations of existing risk assessment methods with the capabilities and limitations of the HEFFR framework

Category	Subcategory	Risk Assessment Methods	Component Failures	Human Errors	Human Errors, Component Failures Combined	Failure Propagation	Early Design Stage	Dynamic Simulations	Hardware-Software Interaction Failures	Software-Human Interaction Failures	Subjective
Human Reliability Assessment	Non-cognition Focused	THERP [34]	No	Yes	No	Yes	No	Yes	No	Yes	Yes
		SHERPA [36] , SPAR-H [38]	No	Yes	No	No	No	No	No	Yes	Yes
		HEART [25, 43–49]	No	Yes	No	No	No	No	No	Yes	No
		SLIM [37, 50, 51]	No	Yes	No	No	No	No	No	Yes	No
	Cognition Focused	HCR [35], ATHEANA [39], MERMOS [41, 42]	No	Yes	No	No	No	No	No	Yes	Yes
		CREAM [40, 52–56, 61, 62]	No	Yes	No	No	No	Yes	No	Yes	No
	Dynamic Simulation Methods	PSPHERE [63], Ref. [60]	No	Yes	No	No	No	Yes	No	Yes	No
	Early Design	TECHR [58], THEA [57]	No	Yes	No	No	Yes	No	No	Yes	Yes
		eMHRA [59]	No	Yes	No	Yes	No	No	No	Yes	Yes
Component Failure Assessment	Traditional Methods	FMEA [64, 67–72]	Yes	Yes	No	No	No	No	No	No	No
		FTA [65, 73–77]	Yes	Yes	No	No	No	Yes	No	No	No
		ETA [66, 78–80]	Yes	Yes	No	Yes	No	No	No	No	Yes
		RBD [82], Bow-tie diagrams [81]	Yes	No	No	Yes	No	No	No	No	Yes
	Early Design Stage	FFDM [83, 84]	Yes	No	No	No	Yes	No	No	No	No
		CSCSIT [85]	Yes	No	No	No	Yes	No	No	No	No
		COBRA [86]	Yes	No	No	Yes	Yes	No	No	No	No
		FFIP [23, 90–92]	Yes	No	No	Yes	Yes	Yes	Yes	No	No
		Bayesian Network [88, 89]	Yes	No	No	Yes	Yes	Yes	No	No	Yes
	Hardware-Software Interaction	Markov Chain-based [93, 94]	Yes	No	No	Yes	No	Yes	Yes	No	Yes
		Environmental Modeling-based [95–97]	Yes	No	No	Yes	Yes	Yes	Yes	No	No
Systemic Risk Assessment		STAMP [98], FRAM [99], Ref. [100]	Yes (organizational level)	Yes (organizational level)	Yes (organizational level)	Yes	No	No	Yes	No	Yes
Human Error + Component Failures		FHEDM [108], Ref. [107]	Yes	Yes	Yes	No	Yes	No	No	No	No
		HEFFR	Yes	Yes	Yes	Yes	Yes	Yes (Event Time or Discrete Time)	Not validated	Not Validated	Minimal

5.4.3 Discussion

This research has validated the HEFFR framework in terms of its ability to predict failures realistically and its comparability with existing risk assessment methods. The results show that the HEFFR framework generated all the severe train accident scenarios and assigned appropriate severity to these failures. Moreover, the module behaviors, component behaviors, and human actions that were most common in severe train accidents were also easily identifiable. In addition to generating the accident scenarios, the HEFFR framework also generated scenarios that would have the same outcome but with different event sequences, allowing designers to identify low probability high impact scenarios that otherwise would have had low expected costs (due to the lower probabilities). Also, the accident scenarios had shorter paths to failures when compared with a majority of the potential fault scenarios. However, the HEFFR scenarios were not able to represent the minute details that were present in the accident scenarios because of the discrete nature of the simulation and early design application of the HEFFR framework. The comparison study showed that the HEFFR framework cannot replace the existing high fidelity risk assessment methods but can complement them by catching potential failures early in the design stages and minimizing the chances of finding major vulnerabilities when more detailed studies are performed.

The validation study also showed that the HEFFR framework can help designers mitigate severe failure through multiple fronts. The straightforward way to identifying potential worst-case fault scenarios is to prioritize scenarios based on expected cost. Another approach will be to analyze the failures with the shortest event sequence because the validation study showed that all of the accident scenarios had a shorter time to failure. One may also identify the specific system behaviors that can contribute to worst outcomes by taking the cumulative expected cost of module behaviors based on the final event, identifying the specific component behaviors that had the highest contribution to the module behaviors with the highest expected costs, and pinpointing human actions that were present in most human action combinations that resulted in the component behaviors with the highest contribution. If such behaviors are identified early on, designers may include design strategies that prevent the system from going into those behaviors. The HEFFR framework does not include any

data synthesis. Rather, the focus is to provide as much data as possible to the designers so they can tailor the data synthesis to the needs and specifications. If the designers are aiming to mitigate the most severe failures, the data synthesis approach used in this study will be ideal. However, if the purpose of assessment is different (e.g., risk-based trade studies), the data synthesis approach will have to be tailored to fit that specific purpose (e.g., calculate the overall system's expected costs and compare with other design alternatives).

The comparison study with existing risk assessment methods showed that the HEFFR framework is capable of modeling the combined effects of human errors and components failure during early design stages (a trait no other method had). However, there were areas that HEFFR could not achieve while other methods could (e.g., higher accuracy of the risk matrices, continuous-time dynamic simulations, fidelity levels, etc.). Hence, the HEFFR framework cannot be used to replace existing risk assessment methods. Instead, it should be used to complement them. The HEFFR framework should be used to analyze (and mitigate) risk early on and inform the later design stage detailed assessments. Since the HEFFR framework lacks fidelity and may be subject to uncertainties, some finer details may go unnoticed. However, if the risk insights are used to inform the late design stage detailed risk assessments such details may come to light. Also, such usage of the HEFFR framework will minimize the chances of finding major vulnerabilities later in the design stages. In summary, when used to complement the existing risk assessment methods, HEFFR can enhance the capabilities of the overall risk assessment and improve safety, performance, cost, and efficiency.

The lack of fidelity of the HEFFR framework fault scenarios is expected because it is an early design stage risk assessment tool. During conceptualization, there are only minimal system data available. The goal of the HEFFR framework is to generate fault scenarios with the minimal data available to help designers identify potential risks with enough fidelity. The validation study showed that the HEFFR framework can achieve this even when the fault scenarios are low fidelity. The HEFFR framework does not account for any uncertainties during fault modeling and risk quantification. There are higher uncertainties during the early design stages. While the results from this study prove that the HEFFR framework can help designers identify and prioritize worst-case fault scenarios,

there is no validation on how the uncertainties can affect risk quantification and prioritization. Since the HEFFR framework is an early design stage risk assessment method, it is important to study the effects of uncertainties on the overall risk model. Future work should explore and validate the effects of uncertainties in the HEFFR framework's risk model.

5.5 Applying the HEFFR Framework to Perform Risk-informed Ergonomic Assessments

This section aims to demonstrate an application of the HEFFR framework to perform risk-informed ergonomic assessments. We couple the HEFFR framework with Digital Human Modeling (DHM) simulations to analyze physical ergonomics and visualize human product interactions. The HEFFR framework is used to identify and prioritize human actions with the worst outcomes. DHM is used to analyze the ergonomics surrounding the worst-case human actions. When the HEFFR framework is applied alone, it is not capable of analyzing ergonomic vulnerabilities. Without the HEFFR framework, designers have no way of prioritizing the needed ergonomic assessments. Hence, coupling the HEFFR framework with DHM complements each other and enhances their capabilities.

5.5.1 Methodology

As shown in Fig. 5.7, the first step is to generate a system model and perform a HEFFR simulation. Next, the resulting data is synthesized to identify and prioritize worst-case human actions. Note that this research does not recommend any specific data synthesis approaches, allowing designers to tailor the data synthesis based on their goals. For example, if the goal is to identify the human actions that can have the greatest reduction in the probability of failures, designers may synthesize the data similar to the approach presented in section 4.5. If the goal is to identify the human actions that contribute to the most severe failures, designers may analyze the data using the approach presented in section 5.4. Once the worst-case human actions are identified, designers may use traditional

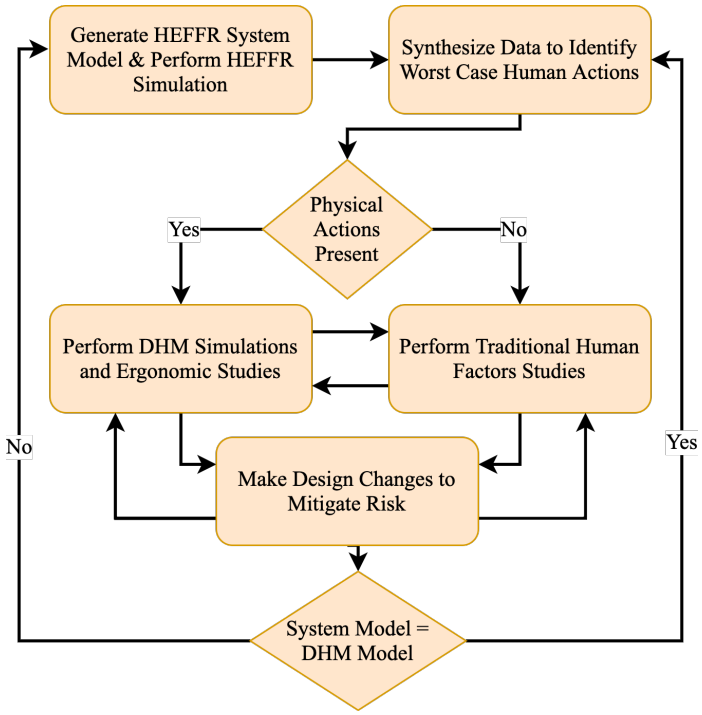


Figure 5.7: Workflow of Performing Risk Informed Ergonomic Assessments Using the HEFFR Framework

human factors approaches to analyze ergonomics and mitigate risk if the actions are cognitive. If the identified human actions are physical, they may use DHM to analyze ergonomics and mitigate risk. They may switch back and forth between the DHM assessments and human factors approaches as they see fit.

To perform DHM simulations, first, a low fidelity Computer-Aided Design (CAD) model is created and loaded into a DHM tool (e.g., Siemens Jack). Then, for each of the identified worst-case human actions (that are physical), the desired ergonomic studies are performed. For example, for the action reacR, a designer may choose to use the reach envelope analysis, reach obscuration, or if the reach requires severe postural changes, such as asymmetric bending, a comfort analysis. DHM has a wide range of potential ergonomic assessment tools to choose from. Examples of DHM analysis tools include National Institute for Occupational Safety and Health (NIOSH), Lifting Index and Rapid Upper Limb Assessment (RULA), and vision analysis and obscuration zone evaluations. In addition,

using DHM can also help designers perform more complex early design stage computational studies such as exploring the design space using surrogate models to optimize human performance [221] and combining DHM with virtual reality and motion capture technology to perform human subject data collection [222]. Designers may choose the best form of analysis depending on the problem they are solving, the worst-case human actions, and the available data.

Based on the analysis results, designers may chose to make design changes, suggest training, develop safe operating procedures and safety protocols, or run further assessments. If the modified design requires changes to the system model, the system model is updated to reflect the changes. Then, a new HEFFR analysis is performed. This process may be iterated until a satisfactory design is derived.

5.5.2 Case Study: Train Locomotive Design

The same train locomotive design case study used in the previous sections in this chapter is used to demonstrate the application of the HEFFR framework to perform risk-informed ergonomic assessments. To identify and prioritize the worst-case human actions, the same data synthesis approach used in section 5.4 is used (i.e., the human actions that had the highest contribution to the worst-case fault scenarios are identified). First, the worst-case module behaviors are identified by calculating the cumulative costs of module behaviors based on if they are present in the last time step. Then, for each module, the module behavior with the highest cumulative cost is identified. Then, the component behaviors are ranked based on their contribution to each module behavior with the highest cumulative cost. Then, human actions are ranked based on their contribution to the highest-ranked component behavior in each module. The contribution of human actions is calculated by considering the number of action combinations (with the specific human actions) that can result in the highest-ranked human induced component behaviors.

This research uses Siemens Jack as the DHM tool to model the train locomotive. Jack has various human performance capabilities such as vision and reach envelop assessments, hand clearance and

interference assessment, force-influenced posture prediction, etc., and multiple ergonomic analysis tools such as static and real-time fatigue, low back analysis, NIOSH, OWAS, RULA, static strength predictions, time standards, etc [223]. It also has occupant packaging tools which include comfort assessment, SAE packaging guidelines, multiple vision zones, etc. [223]. For this study, a low-fidelity CAD model of a diesel-electric train locomotive is developed and injected into the DHM platform to perform ergonomic assessments. The results of this application are discussed in the following section.

5.5.3 Results

In section 5.4, it was found that the module behaviors "failed," "failed throttle on," and "failed no braking" from modules Wheel Assembly, Throttle and Dynamic Brake System, and Air Brake System respectively, had the highest cumulative costs for each respective modules. To mitigate the risk of the Wheel Assembly failing, designers may choose to install a health monitoring system and establish safety protocols to mitigate severity when failures are detected. The system model can be updated, and the scenarios involving the safety protocol can be simulated using HEFFR to check the effectiveness of the protocols. The human induced behaviors "failed close" and "failed throttle on" of the components brake valve and throttle lever from the modules Air Brake System and Throttle and Dynamic Brake System respectively, were the highest-ranked human induced component behaviors for each of the modules. As shown in Fig. 5.8, the human actions for the brake valve can be ranked as Turn, Detect, Reach, See, and Grasp based on the contribution of each human action to the behavior "failed close." Similarly, the human actions for the throttle lever can be ranked as Detect, Move, Reach, Grasp, and See. Designers may prioritize ergonomic assessments for each of these components based on the human action ranking. Among the components, the throttle lever should be given higher priority because the module behavior relating to the Throttle and Dynamic Brake had the highest cumulative expected cost.

Action Detect is cognitive, and action See is both cognitive and physical. Also, from the action sequence graphs, the successful execution of the action See is a precursor for the success of action

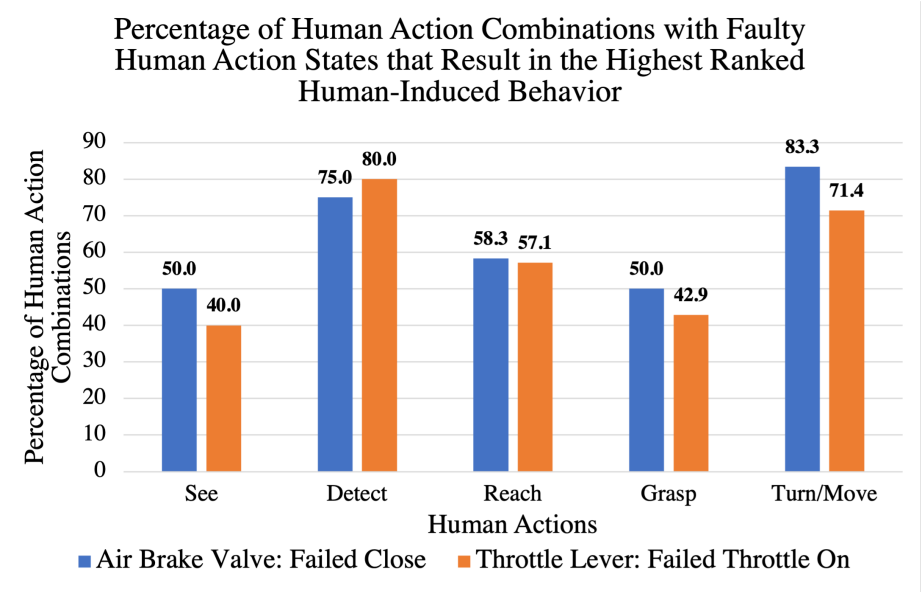
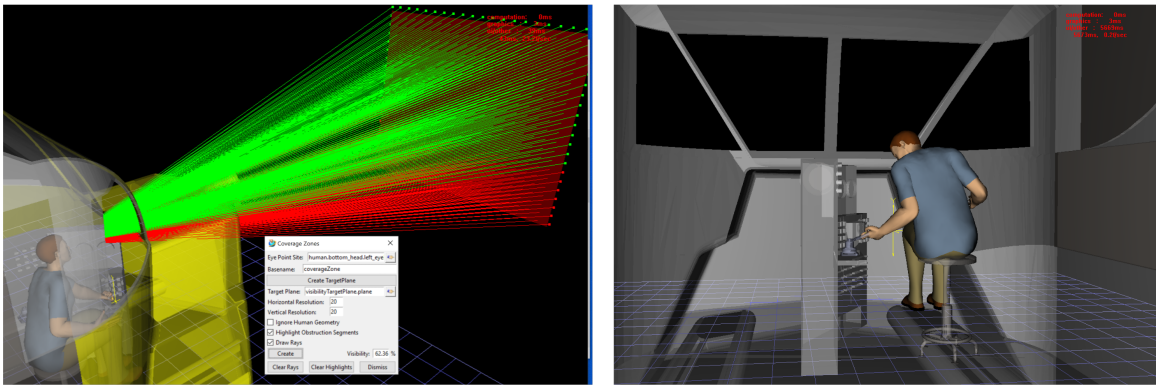


Figure 5.8: The percentage of human action combinations with faulty human action states that result in the highest ranked human induced behavior

Detect (i.e., if the operator fails to see, action Detect all always fail). Since the action Detect is among the top-ranked for both components (especially for the throttle lever), designers may prioritize the actions Detect and See. To improve the action Detect, they may suggest training, improve signal salience, or add redundant signals. To improve the cognitive aspects of the action See, designers may suggest training or explore ways to keep the operator's attention at appropriate levels to promote a higher chance of detection. The rest of the actions are physical and are assessed using DHM. While the actions Turn and Move for the brake valve and throttle lever were ranked the highest among the physical actions, the actions See, Reach, and Grasp cannot be ignored because from the action sequence graphs, we know that the success of these actions is a requirement for the actions Turn and Move to be successful (i.e., one cannot turn the valve without reaching and grasping it). However, the actions for the throttle lever should get a higher priority than the brake valve because the module behavior relating to the throttle lever had the highest cumulative expected cost.

Based on the prioritization, reach- and vision-related assessments were identified as the necessary DHM assessments. Vision-related assessments were chosen because of their relation to the action



Vision Coverage Assessment while Reaching the Throttle Lever

Reach Posture while Reaching the Brake Valve

Figure 5.9: Reach postures and vision obcuration zones (only for while reaching the throttle lever of the 95^{th} percentile U.S. Male when reaching the throttle lever and brake valve

Detect. Reach-related assessments were chosen to analyze the actions Reach, Grasp, and Turn. The DHM assessments were performed for both 5^{th} percentile U.S. female manikin and 95^{th} percentile U.S. male from the ANSUR anthropometric library to make sure that analysis covers the needs of the majority of the population. As shown in Fig. 5.9, the 95^{th} percentile U.S. male had to bend to reach the throttle lever and brake valve. Also, the vision obscuration zones were analyzed while reaching the throttle lever because this will be the default posture of the train conductor. The vision obscuration assessment showed that there is 37 percent obscuration when the U.S. 95^{th} percentile U.S. male manikin was reaching the throttle lever. Similar to the 95^{th} percentile U.S. male, the 5^{th} percentile U.S. female manikin also had to bend to reach the throttle lever and brake valve but to a lesser degree. The vision obscuration was worse for the 5^{th} percentile U.S. female.

From the DHM assessments, it becomes apparent that the placement of the throttle lever and brake valve need to be moved because both manikins had to bend to reach them. Bending for long periods is associated with lower back injuries. To improve the placement of the controls, designers may relocate them or change the seating position. During this process, they may not be able to find an optimal position that promotes maximum vision coverage while having no bending postures when reaching the throttle lever and brake valve. In such a case, since the brake valve had lesser priority than the throttle lever and the train conductor will only be reaching the brake valve occasionally,

they may change the placement of the throttle lever to make sure that there is no bending while allowing minimal bending for reaching the air brake. Regardless of the placement of the throttle lever and brake valve, there should not be any comprise in the vision coverage because of the important role it plays with detection. Designers should make an effort to maximize the vision coverage. These adjustments can be done manually by moving the manikin and the controls in the DHM environment, or a surrogate model-based design exploration can be performed using the framework proposed by Ahmed et al. [221].

5.5.4 Discussion

In this case study, we present an application of the HEFFR framework to perform risk-informed ergonomic analysis during early design stages. Also, the proposed application allows designers to visualize human product interactions. The results above indicate how the HEFFR framework's results can be used to prioritize ergonomic assessments. The results also show that such prioritization can be helpful in terms of finding the right ergonomic assessments and when design trade-offs are required. Being able to prioritize human actions and perform ergonomic assessments early in design will allow designers to make design decisions based on ergonomics, which would potentially help to improve the overall performance and safety of the system. Also, when trade-offs are needed between ergonomic features, they may make informed decisions based on risk-based prioritization. In addition, designers can apply the HEFFR framework along with DHM to perform more complex analysis with motion capture and virtual reality systems to have a more exhaustive risk-based design space exploration without having to rely on fully functional physical prototypes. Obtaining data about potential human-product interaction errors early in the design can also contribute to the efforts for reducing the overall design cost.

The application of the HEFFR framework for risk-based ergonomic assessment is one of many applications of the HEFFR framework. When applied for risk-based ergonomic assessments, the HEFFR framework complements the traditional ergonomic assessment process by allowing the pri-

oritization of ergonomic assessments and ergonomic features based on risk. Also, this application enhances the capabilities of the HEFFR framework by allowing the assessment of ergonomic vulnerabilities. Some examples of other potential applications of the HEFFR framework are risk-based trade-off studies, component selection studies, resilient design, establishing operating procedures and safety protocols, and training development.

5.6 Conclusion

We have demonstrated the applicability of the HEFFR framework to complex engineered systems, validated the framework in terms of its ability to predict failures, demonstrated its application to perform risk-informed ergonomic assessments. To study the applicability of the HEFFR framework to complex engineered systems, we first introduced a modular analysis approach to manage the complexity and computational cost of performing HEFFR assessments on complex engineered systems. Then, we validated the modular analysis approach to ensure that it produces consistent results when compared to integral assessments. We also explored the consistency surrounding different module partitionings. The study shows that the proposed modular analysis approach can produce consistent results for the modules assessed, justifying the use of modular HEFFR assessments for a train locomotive design study. When performing modular assessments, the data analyses need to be tailored with the modular fault simulation in mind to extract similar information to integral assessments.

The validation study had two parts. The first part explored the ability of the HEFFR framework to predict and prioritize failures realistically. To achieve this, events that led to past severe train accidents were converted to HEFFR fault scenario form and compared with the HEFFR results of a train locomotive design case study. The results showed that the HEFFR framework generated all of the scenarios that led to the past severe accidents and assigned appropriate serveries to all of them. In addition, The HEFFR framework was able to pinpoint the module behaviors, component behaviors, and human actions that were most common in the accidents. The second part of the validation

study compared the HEFFR framework with existing risk assessment methods. The capabilities and limitations of existing risk assessment methods were identified from the literature and compared against the merits and limitations of the HEFFR framework to define the ideal usage of the HEFFR framework. The HEFFR is best used when applied during early design stages to complement the late design stage risk assessment process. Such an implementation will minimize the chances of finding major risks later on when design changes can introduce more vulnerabilities and increase cost and time-to-market.

Finally, an application of the HEFFR framework was demonstrated. The HEFFR framework was applied to perform risk-informed ergonomic assessments. The results from the HEFFR framework were used to prioritize ergonomic assessments based on their contribution to overall system risk. Then, the use of DHM to perform ergonomic assessments and inform design was demonstrated using the train locomotive design case study. The prioritization of ergonomic assessments can help designers choose the correct ergonomic studies and inform design decisions when trade-offs are needed between ergonomic features.

Chapter 6: Conclusions

The goal of this research is to enable designers to assess the effects of human- and component-related vulnerabilities during early design stages. Achieving this goal will minimize late design stage redesign and improve performance, safety, costs, and time-to-market. To achieve this goal, this work has introduced an early design stage computational risk assessment framework (Human Error and Functional Failure Reasoning (HEFFR)) to assess the system-level effects of human errors and component failures acting together (and in isolation). The HEFFR framework assesses risk using three main components; scenario generation, fault prediction, and risk quantification. The fault prediction model is developed by expanding the Functional Failure Identification and Propagation (FFIP) [24] framework to include the human aspects of the system. Then, the fault prediction model is validated using the Air France 447 crash and a flight simulator. The automated scenario generated approach uses a modified depth first search to generate a wide range of potential fault scenarios involving both components and humans. The risk quantification model quantifies the expected cost of failures and likelihood of occurrence using principles from reliability engineering, human factors, and resiliency modeling.

The overall framework's applicability to complex engineered systems is studied by introducing a HEFFR based modular risk assessment approach to manage complexity. The study showed that the modular risk assessment approach yielded results similar to integral assessment regardless of the number of partitions and reduced the computation costs significantly. The validity of the HEFFR framework is studied on two fronts. First, past train accidents with severe outcomes were compared with the results of the HEFFR analysis of a train locomotive design. The results showed that the HEFFR framework generated all the scenarios that led to the accidents and assigned severities appropriately. In addition to identifying the most severe fault scenarios, the HEFFR framework was also able to pinpoint rarer fault scenarios with similar outcomes to the accident scenarios (scenarios

with lower likelihood but high failure cost). Next, the HEFFR framework is compared with existing risk assessment methods to understand its usage. Results show that the HEFFR framework works best when it is used to complement more detailed risk assessment methods. Finally, the application of the HEFFR framework to inform risk-based ergonomic assessments is demonstrated. The results from the HEFFR framework are used to identify and prioritize human actions with the worst outcomes to perform ergonomic assessments using digital human modeling.

The next sections will conclude this dissertation by discussing the contributions and implications of this research and pointing out limitations that may be addressed in future work.

6.1 Contributions and Implications

This work has numerous contributions to the field of risk-based design. First, it will enable the assessment of risks relating to human errors and component failures in combination rather than in isolation during early design stages. Conventional risk assessment techniques either assess component failures or human errors in isolation, are not capable of assessing the propagation of failures and/or are not applicable during early design stages. This work overcomes these limitations by introducing a discrete time-based fault propagation modeling framework that can assess the combined risk of component failures and human errors during early design stages. The second contribution is the introduction of the automated scenario generation and prioritization process that appropriately utilizes computational resources to express sufficient model fidelity using the minimal data available during early design stages. The automated fault scenario generation and prioritization allow the identification and consideration of worst-case fault scenarios early in the design, enabling the incorporation of mitigating strategies early on. Also, it allows designers identify human errors that do not cause immediate harm to the system and may easily be overlooked. This is important because such underlying errors may easily turn into catastrophic failure when they interact with other elements of the system. Lastly, this work demonstrated an approach to perform ergonomic assessments and visualize human-product interactions before any physical prototypes are built. No early design stage

human-centered digital prototyping framework exists that can inform designers on the ergonomic priorities when it comes to promoting optimal system performance and safety. They usually rely on experts or are realized later in the design process when more details about the design are present [224]. This research demonstrated an application of HEFFR framework that injected digital human modeling, fault modeling, and reliability analysis strategies into a computational design environment to prioritize ergonomic assessments to minimize system vulnerabilities earlier in the design process with the goal of improving overall system safety and performances.

Setting up the system model and creating the behavior models and human action models for the HEFFR framework can help designers think deeper about the system under design, resulting in well-thought-out designs. The framework can also be used to help design decision-making during the early design stages. For example, it may be used to perform trade studies to evaluate alternate designs. The HEFFR framework can also be used to identify points of automation and points of human intervention based on risk. With the identification of component behavior-related risks, the HEFFR framework can inform component selection. The overall framework can be used to identify safe operating procedures and inform safety protocols. For example, safety protocols can be developed and tested for worst-case scenarios so that operators may identify them early and take preventive measures. It can also inform training in terms of helping with failure prevention. Since the HEFFR framework takes the performing shaping factors when calculating severities, it can help designers understand operator requirements (e.g., experience level, workload limits, etc.) for safe operation. When coupled with digital human modeling, the HEFFR framework can inform occupant packaging and interface layout design. In summary, the HEFFR framework can help designers make risk-informed decisions regarding both component- and human-related elements in the system during early design stages.

With the advancements in artificial intelligence and the shift towards automation, how humans interact with systems has to be redefined. Regardless of the industry (e.g., automobile, manufacturing, aviation, etc.), the human's role (as an operator, maintainer, or end-user) needs to be re-assessed. Some of the challenges surrounding this can be resolved by exploring questions such as which

functions to automate? In what circumstances does the human need to take control? What happens when a failure occurs? Even in a fully automated system such as a self-driving car, humans still need to be kept in the loop to make sure that they can take over when situations demand it. To successfully address these challenges, designers need to consider both human- and machine-elements concurrently and explore solutions that minimize risk and improve safety and human well-being. The HEFFR framework can help designers explore these questions early in the design stages. For example, designers may identify when humans need to take control by identifying events that can lead to the worst outcomes and studying mitigation actions that can prevent the system from further cascading into the worst outcome.

The HEFFR framework can promote more sustainable products through resource conservation and failure prevention. When design changes are made late in the design stages, they are costly and time-consuming. This often required extra resources and time (e.g., budget overruns and missed deadlines of the F-35 joint striker program [225]). By moving risk assessment to early design stages, the HEFFR framework helps designers identify potential risks early on, minimizing the chances for design changes later. As a result, the additional resources spent on late design stage design changes can be minimized. When accidents occur in complex engineered systems, they can have lasting effects on the economies, societies, and environments these systems operate on (e.g., the effects of Bhopal gas leak were still present after 20 years [14]). By mitigating the potential failures, the HEFFR framework can minimize the effects of failures on the economies, societies, and environments complex systems operate on. Updating detailed simulation models when new elements are introduced to the systems can be expensive and resource-intensive. Since the HEFFR simulations are relatively easy to set up and computationally inexpensive, the HEFFR system model can act as a low fidelity digital twin to test upgrades, change in operation procedure, and new safety protocols when the system is in use. While it cannot replace the more detailed evaluations and testing, it can buy designers time by allowing quick and easy simulations until more detailed simulation models are available and reduce the need for repetition by reducing the chances of having to update the detailed models often. As a result, the overall cost can be minimized while promoting more sustainable products.

Going back to the Boeing 737 Max debacle discussed in Chapter 1, let us imagine that the engineers at Boeing had access to a tool like HEFFR. They may have been able to identify the risk of their design changes with the engine and MCAS system early in the design stages. They would have been more likely to address those risks rather than ignoring them because the cost and time pressure would have been minimal. As a result, the loss of life and the economic impact could have been avoided. All these are speculations, and we may never know the course of action Boeing would have taken unless the events are repeated with the availability of a tool like HEFFR. The question of if the HEFFR framework would have been able to capture the risks that were present with the Boeing 737 Max can only be answered if the aircraft is modeled using HEFFR. The goal of this research is to not answer that question but to take a step in that direction, in making a tool available for designers to assess the combined effects of failures as captain Sullenberger put it–*"Each aircraft manufacturer must have a comprehensive safety risk assessment system that can review an entire aircraft design holistically, looking for risks, not only singly, but in combination [6]."*

6.2 Future Work

Chapters 3, 4, and 5 in this dissertation discuss potential future work within the context of the research presented in them. This section will explore potential future work opportunities that are not covered in those chapters. The HEFFR framework does not consider uncertainties when modeling faults or quantifying severity. Since the fault modeling relies on system models that are created by experts, the system representations and the behavior models can vary from expert to expert due to the abstract nature of the system details available during early design stages. Similarly, the uncertainties present in the variables (e.g., failure rates, failure distributions, human error producing condition factors, etc.) in the risk quantification model can affect the severity quantification. Future work should explore how to account for uncertainties in the HEFFR framework to enable designers to best account for it in hazard modeling and risk-based decision-making. Another area of future work can explore how the HEFFR framework compares with existing risk assessment methods in terms

of its ability to predict failures. The validation study presented in this research does not explore if the HEFFR framework generates more or less risk insights when compared to existing methods. A study with experts applying the HEFFR framework along with other risk assessment methods to the design of the same system will help identify if the HEFFR framework is capable of generating similar risk insights to existing risk assessment methods.

Another area of future work may explore the automation of fault modeling using the HEFFR framework. Past research has explored automating functional modeling using a design repository and machine learning [226]. A similar approach can be explored to creating the system representation of the HEFFR framework automatically. Components such as behavior models and action classifications in the HEFFR framework are repeatable, meaning that the behavior modeling and simulations can be automated. With the availability of historic accident data, machine learning can be employed to study failure patterns and predict failures smartly. On the whole, future work may examine design repositories and past accident data with machine learning to inform automated system modeling and fault prediction using the HEFFR framework. The automation of risk assessment will bring consistency and minimize subjectivity while making them faster and more efficient. The HEFFR framework is only capable of modeling humans in an enclosed space (e.g., a pilot in a cockpit). It is not capable of modeling human interactions that occur from outside of an enclosed space, making it useless when modeling systems where a human is not in an enclosed space (e.g., a fleet of delivery drones operated through a control tower). With the move towards Industry 4.0 and advances in autonomous systems, it is important to be able to analyze the human interactions with such systems. Future work may expand the HEFFR framework to unlock the modeling of human interactions is systems that involve human interactions outside of an enclosed space.

Bibliography

[1] J Herkert, J Borenstein, and K Miller, "The boeing 737 max: lessons for engineering ethics", Science and engineering ethics **26**, 2957–2974 (2020).

[2] KNKT, *Final aircraft accident investigation report: pt. lion mentari airlines*, Investigation Report KNKT.18.10.35.04 (Komite Nasional Keselamatan Transportasi, Republic of Indonesia, Tanjung Karawang, West Java, Oct. 2019).

[3] AAIB, *Interim Investigation Report on Accident to the B737-8 (MAX) Registered ET-AVJ Operated by Ethiopian Airlines*, Investigation Report AI-01/19 (Aircraft Accident Investigation Bureau, Ministry of Transport, The Federal Democratic Republic of Ethiopia, Mar. 2020).

[4] FAA, *Faa updates on boeing 737 max*, On FAA.gov, https://www.faa.gov/news/updates/?newsId=93206.

[5] C Isidore, *Boeing's 737 max debacle could be the most expensive corporate blunder ever*, On CNN Business, URL: https://www.cnn.com/2020/11/17/business/boeing-737-max-grounding-cost/index.html, Nov. 2020.

[6] CB Sullenberger III, *Prepared statement of chesley b. "sully" sullenberger iii*, Hearing titled, "Status of the Boeing 737 MAX: Stakeholder Perspectives," Subcommittee on Aviation of the Committee on Transportation and Infrastructure, U.S. House of Representatives, 116th Congress, First Session, https://transportation.house.gov/imo/media/doc/Sully% 20Sullenberger%20Testimony.pdf, June 2019.

[7] MS Donaldson, JM Corrigan, LT Kohn, et al., *To err is human: building a safer health system*, Vol. 6 (National Academies Press, 2000).

[8] L Högberg, "Root causes and impacts of severe accidents at large nuclear power plants", Ambio **42**, 267–284 (2013).

[9] SA Shappell and DA Wiegmann, "Us naval aviation mishaps, 1977-92: differences between single-and dual-piloted aircraft.", Aviation, Space, and Environmental Medicine **67**, 65–69 (1996).

[10] DA Wiegmann and SA Shappell, "Human error analysis of commercial aviation accidents: application of the human factors analysis and classification system (hfacs)", Aviation, space, and environmental medicine **72**, 1006–1016 (2001).

[11] J Reason, "The contribution of latent human failures to the breakdown of complex systems", Philosophical Transactions of the Royal Society of London B: Biological Sciences **327**, 475–484 (1990).

[12] A Sneddon, K Mearns, and R Flin, "Situation awareness and safety in offshore drill crews", Cognition, Technology & Work **8**, 255–267 (2006).

[13] N Meshkati, "Human factors in large-scale technological systems' accidents: three mile island, bhopal, chernobyl", Industrial Crisis Quarterly **5**, 133–154 (1991).

[14] E Broughton, "The bhopal disaster and its aftermath: a review", Environmental Health **4**, 6 (2005).

[15] HO Demirel, "Modular human-in-the-loop design framework based on human factors", PhD thesis (Purdue University, 2015).

[16] D Norman, *The design of everyday things: revised and expanded edition* (Constellation, 2013).

[17] DG Ullman, *The mechanical design process*, Vol. 2 (McGraw-Hill New York, 1992).

[18] H Walsh, A Dong, and I Tumer, "Towards a theory for unintended consequences in engineering design", in Proceedings of the design society: international conference on engineering design, Vol. 1, 1 (Cambridge University Press, 2019), pp. 3411–3420.

[19] HO Demirel and VG Duffy, "Applications of digital human modeling in industry", in International conference on digital human modeling (Springer, 2007), pp. 824–832.

[20] HO Demirel and VG Duffy, "Digital human modeling for product lifecycle management", in International conference on digital human modeling (Springer, 2007), pp. 372–381.

[21] J Dul, R Bruder, P Buckle, P Carayon, P Falzon, WS Marras, JR Wilson, and B van der Doelen, "A strategy for human factors/ergonomics: developing the discipline and profession", Ergonomics **55**, 377–395 (2012).

[22] N King and A Majchrzak, "Concurrent engineering tools: are the human issues being ignored?", IEEE Transactions on engineering management **43**, 189–201 (1996).

[23] T Kurtoglu and IY Tumer, "A graph-based fault identification and propagation framework for functional design of complex systems", Journal of Mechanical Design **130**, 051401 (2008).

[24] T Kurtoglu and IY Tumer, "Ffip: a framework for early assessment of functional failures in complex systems", ICED, Cite des Sciences et de L'industrie, Paris, France (2007).

[25] J Williams, "A data-based method for assessing and reducing human error to improve operational performance", in Human factors and power plants, 1988., conference record for 1988 ieee fourth conference on (IEEE, 1988), pp. 436–450.

[26] D Hulse, C Hoyle, K Goebel, and IY Tumer, "Quantifying the resilience-informed scenario cost sum: a value-driven design approach for functional hazard assessment", Journal of Mechanical Design **141**, 021403 (2019).

[27] A Joshi, M Whalen, and M Heimdahl, "Model-based safety analysis final report", NASA Techreport (2005).

[28] M Stamatelatos, H Dezfuli, G Apostolakis, C Everline, S Guarro, D Mathias, A Mosleh, T Paulos, D Riha, C Smith, et al., "Probabilistic risk assessment procedures guide for nasa managers and practitioners", (2011).

[29] M Lyons, S Adams, M Woloshynowych, and C Vincent, "Human reliability analysis in healthcare: a review of techniques", International Journal of Risk & Safety in Medicine **16**, 223–237 (2004).

[30] B Kirwan, *A guide to practical human reliability assessment* (CRC press, 1994).

[31] J Deeter and E Rantanen, "Human reliability analysis in healthcare", in Proceedings of symposium on human factors and ergonomics in health care (2012), pp. 45–51.

[32] B Kirwan, "Human error identification techniques for risk assessment of high risk systems—part 1: review and evaluation of techniques", Applied ergonomics **29**, 157–177 (1998).

[33] NA Stanton and SV Stevenage, "Learning to predict human error: issues of acceptability, reliability and validity", Ergonomics **41**, 1737–1756 (1998).

[34] A Swain, "Therp technique for human error rate prediction", in Proceedings of the symposium on quantification of human performance, albuquerque (1964).

[35] G Apostolakis, V Bier, and A Mosleh, "A critique of recent models for human error rate assessment", Reliability Engineering & System Safety **22**, 201–217 (1988).

[36] D Embrey, "Sherpa: a systematic human error reduction and prediction approach", in Proceedings of the international topical meeting on advances in human factors in nuclear power systems (1986).

[37] D Embrey, P Humphreys, E Rosa, B Kirwan, and K Rea, *Slim-maud: an approach to assessing human error probabilities using structured expert judgment. volume i. overview of slim-maud*, tech. rep. (Brookhaven National Lab., Upton, NY (USA), 1984).

[38] D Gertman, H Blackman, J Marble, J Byers, C Smith, et al., "The spar-h human reliability analysis method", US Nuclear Regulatory Commission (2005).

[39] SE Cooper, A Ramey-Smith, J Wreathall, and G Parry, *A technique for human error analysis (atheana)*, tech. rep. (Nuclear Regulatory Commission, Washington, DC (United States). Div. of Systems Technology; Brookhaven National Lab., Upton, NY (United States); Science Applications International Corp., Reston, VA (United States); NUS Corp., Gaithersburg, MD (United States), 1996).

[40] E Hollnagel, "Cream-cognitive reliability and error analysis method", in (1998).

[41] P Le Bot, F Cara, and C Bieder, "Mermos, a second generation hra method: what it does and doesn't do", in Proceedings of the international topical meeting on probabilistic safety assessment (psa'99), Vol. 2 (1999), pp. 852–880.

[42] P Meyer, P Le Bot, and H Pesme, "Mermos: an extended second generation hra method", in 2007 ieee 8th human factors and power plants and hprct 13th annual meeting (IEEE, 2007), pp. 276–283.

[43] B Kirwan and H Gibson, "Cara: a human reliability assessment tool for air traffic safety management—technical basis and preliminary architecture", in *The safety of systems* (Springer, 2007), pp. 197–214.

[44] B Kirwan, H Gibson, R Kennedy, J Edmunds, G Cooksley, and I Umbers, "Nuclear action reliability assessment (nara): a data-based hra tool", in Probabilistic safety assessment and management (Springer, 2004), pp. 1206–1211.

[45] W Gibson, A Mills, S Smith, and B Kirwan, "Railway action reliability assessment, a railway-specific approach to human error quantification", in Proceedings of the australian system safety conference, Vol. 145 (2013), pp. 3–8.

[46] E Akyuz, M Celik, and S Cebi, "A phase of comprehensive research to determine marine-specific epc values in human error assessment and reduction technique", Safety science **87**, 63–75 (2016).

[47] F Castiglia, M Giardina, and E Tomarchio, "Risk analysis using fuzzy set theory of the accidental exposure of medical staff during brachytherapy procedures", Journal of radiological protection **30**, 49 (2010).

[48] Y Guo and Y Sun, "Flight safety assessment based on an integrated human reliability quantification approach", PloS one **15**, e0231391 (2020).

[49] W Wang, X Liu, and Y Qin, "A modified heart method with fanp for human error assessment in high-speed railway dispatching tasks", International Journal of Industrial Ergonomics **67**, 242–258 (2018).

[50] J Tu, W Lin, and Y Lin, "A bayes-slim based methodology for human reliability analysis of lifting operations", International Journal of Industrial Ergonomics **45**, 48–54 (2015).

[51] JL Zhou and Y Lei, "A slim integrated with empirical study and network analysis for human error assessment in the railway driving process", Reliability Engineering & System Safety **204**, 107148 (2020).

[52] Y Guo, Y Sun, X Yang, and Z Wang, "Flight safety assessment based on a modified human reliability quantification method", International Journal of Aerospace Engineering **2019** (2019).

[53] K Yoshimura, T Takemoto, and N Mitomo, "The support for using the cognitive reliability and error analysis method (cream) for marine accident investigation", in 2015 international conference on informatics, electronics & vision (iciev) (IEEE, 2015), pp. 1–4.

[54] J Chen, D Zhou, C Lyu, and X Zhu, "A method of human reliability analysis and quantification for space missions based on a bayesian network and the cognitive reliability and error analysis method", Quality and Reliability Engineering International **34**, 912–927 (2018).

[55] Y Xi, Z Yang, Q Fang, W Chen, and J Wang, "A new hybrid approach to human error probability quantification–applications in maritime operations", Ocean Engineering **138**, 45–54 (2017).

[56] Z Yang, S Bonsall, A Wall, J Wang, and M Usman, "A modified cream to human reliability quantification in marine engineering", Ocean engineering **58**, 293–303 (2013).

[57] S Pocock, MD Harrison, PC Wright, and P Johnson, "Thea: a technique for human error assessment early in design.", in Interact, Vol. 1 (2001), pp. 247–254.

[58] MC Maturana and MR Martins, "Technique for early consideration of human reliability: applying a generic model in an oil tanker operation to study scenarios of collision", Journal of Offshore Mechanics and Arctic Engineering **141** (2019).

[59] N Papakonstantinou, M Porthin, M O'Halloran, and L Van Bossuyt, "A model-driven approach for incorporating human reliability analysis in early emergency operating procedure development", in 2016 annual reliability and maintainability symposium (rams) (IEEE, 2016), pp. 1–6.

[60] A Angelopoulou, K Mykoniatis, and NR Boyapati, "Industry 4.0: the use of simulation for human reliability assessment", Procedia Manufacturing **42**, 296–301 (2020).

[61] S Zhang, W He, D Chen, J Chu, and H Fan, "A dynamic human reliability assessment approach for manned submersibles using pmv-cream", International Journal of Naval Architecture and Ocean Engineering **11**, 782–795 (2019).

[62] M Abbassinia, O Kalatpour, M Motamedzade, A Soltanian, and I Mohammadfam, "Dynamic human error assessment in emergency using fuzzy bayesian cream", Journal of Research in Health Sciences **20**, e00468 (2020).

[63] S Samima and M Sarma, "Psphere: person specific human error estimation", Journal of Reliable Intelligent Environments, 1–32 (2021).

[64] ' US Department of Defense, *Procedures for Performing a Failure Mode, Effects and Criticality Analysis*, Military Standard MIL-STD-1629A (US Department of Defense, Washington DC, 1980).

[65] WE Vesely, FF Goldberg, NH Roberts, and DF Haasl, *Fault tree handbook*, tech. rep. (Nuclear Regulatory Commission Washington dc, 1981).

[66] CA Ericson, "Event tree analysis", Hazard Analysis Techniques for System Safety, 223–234 (2005).

[67] JK Chen, "Prioritization of corrective actions from utility viewpoint in fmea application", Quality and Reliability Engineering International **33**, 883–894 (2017).

[68] GA Keskin and C Özkan, "An alternative evaluation of fmea: fuzzy art algorithm", QUality and reliability engineering international **25**, 647–661 (2009).

[69] K Xu, LC Tang, M Xie, SL Ho, and M Zhu, "Fuzzy assessment of fmea for engine systems", Reliability Engineering & System Safety **75**, 17–29 (2002).

[70] W Song, X Ming, Z Wu, and B Zhu, "A rough topsis approach for failure mode and effects analysis in uncertain environments", Quality and Reliability Engineering International **30**, 473–486 (2014).

[71] HC Liu, JX You, XJ Fan, and QL Lin, "Failure mode and effects analysis using d numbers and grey relational projection method", Expert Systems with Applications **41**, 4670–4679 (2014).

[72] HW Lo, JJ Liou, CN Huang, and YC Chuang, "A novel failure mode and effect analysis model for machine tool risk analysis", Reliability Engineering & System Safety **183**, 173–183 (2019).

[73] L Shi, J Shuai, and K Xu, "Fuzzy fault tree assessment based on improved ahp for fire and explosion accidents for steel oil storage tanks", Journal of hazardous materials **278**, 529–538 (2014).

[74] M Yazdi, F Nikfar, and M Nasrabadi, "Failure probability analysis by employing fuzzy fault tree analysis", International Journal of System Assurance Engineering and Management **8**, 1177–1193 (2017).

[75] H Boudali, P Crouzen, and M Stoelinga, "A compositional semantics for dynamic fault trees in terms of interactive markov chains", in International symposium on automated technology for verification and analysis (Springer, 2007), pp. 441–456.

[76] Y Dutuit and A Rauzy, "Efficient algorithms to assess component and gate importance in fault tree analysis", Reliability Engineering & System Safety **72**, 213–222 (2001).

[77] X Zhang, Q Miao, X Fan, and D Wang, "Dynamic fault tree analysis based on petri nets", in 2009 8th international conference on reliability, maintainability and safety (IEEE, 2009), pp. 138–142.

[78] M Kasaeyan, J Wang, I Jenkinson, and M Miri Lavasani, "Fuzzy consequence modelling of hydrocarbon offshore pipeline", International Journal of MArine Science and Engineering **1**, 3–12 (2011).

[79] D Huang, T Chen, and MJJ Wang, "A fuzzy set approach for event tree analysis", Fuzzy sets and systems **118**, 153–165 (2001).

[80] C Acosta and NO Siu, *Dynamic event tree analysis method (detam) for accident sequence analysis*, tech. rep. (Cambridge, Mass.: Dept. of Nuclear Engineering, Massachusetts Institute of . . ., 1991).

[81] YE Saud, K Israni, and J Goddard, "Bow-tie diagrams in downstream hazard identification and risk assessment", Process Safety Progress **33**, 26–35 (2014).

[82] M Čepin, "Reliability block diagram", in *Assessment of power system reliability* (Springer, 2011), pp. 119–123.

[83] RB Stone, IY Tumer, and M Van Wie, "The function-failure design method", Journal of Mechanical Design **127**, 397–407 (2005).

[84] KG Lough, R Stone, and IY Tumer, "The risk in early design method", Journal of Engineering Design **20**, 155–173 (2009).

[85] Z Huang and Y Jin, "Conceptual stress and conceptual strength for functional design-for-reliability", in Asme 2008 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers, 2008), pp. 437–447.

[86] AR Short, *Design of autonomous systems for survivability through conceptual object-based risk analysis* (Colorado School of Mines, 2016).

[87] D Krus and KG Lough, "Applying function-based failure propagation in conceptual design", in Asme 2007 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers, 2007), pp. 407–420.

[88] M Goswami and M Tiwari, "A predictive risk evaluation framework for modular product concept selection in new product design environment", Journal of Engineering Design **25**, 150–171 (2014).

[89] N Yodo and P Wang, "Resilience allocation for early stage design of complex engineered systems", Journal of Mechanical Design **138**, 091402 (2016).

[90] I Tumer and C Smidts, "Integrated design-stage failure analysis of software-driven hardware systems", IEEE Transactions on Computers **60**, 1072–1084 (2010).

[91] DC Jensen, IY Tumer, and T Kurtoglu, "Modeling the propagation of failures in software driven hardware systems to enable risk-informed design", in Asme international mechanical engineering congress and exposition, Vol. 48777 (2008), pp. 283–293.

[92] S Sierla, I Tumer, N Papakonstantinou, K Koskinen, and D Jensen, "Early integration of safety to the mechatronic system design process by the functional failure identification and propagation framework", Mechatronics **22**, 137–151 (2012).

[93] DS Roy, C Murthy, and DK Mohanta, "Reliability analysis of phasor measurement unit incorporating hardware and software interaction failures", IET Generation, Transmission & Distribution **9**, 164–171 (2014).

[94] X Teng, H Pham, and DR Jeske, "Reliability modeling of hardware and software interactions, and its applications", IEEE Transactions on Reliability **55**, 571–577 (2006).

[95] S Sinha, NK Goyal, and R Mall, "Reliability and availability prediction of embedded systems based on environment modeling and simulation", Simulation Modelling Practice and Theory **108**, 102246 (2021).

[96] MZ Iqbal, A Arcuri, and L Briand, "Environment modeling and simulation for automated testing of soft real-time embedded software", Software & Systems Modeling **14**, 483–524 (2015).

[97] M Auguston, JB Michael, and MT Shing, "Environment behavior models for automation of testing and assessment of system safety", Information and Software Technology **48**, 971–980 (2006).

[98] N Leveson, "A new accident model for engineering safer systems", Safety science **42**, 237–270 (2004).

[99] E Hollnagel, *Fram: the functional resonance analysis method: modelling complex socio-technical systems* (CRC Press, 2017).

[100] NA Stanton and C Harvey, "Beyond human error taxonomies in assessment of risk in sociotechnical systems: a new paradigm with the east 'broken-links' approach", Ergonomics **60**, 221–233 (2017).

[101] K Kazaras, K Kirytopoulos, and A Rentizelas, "Introducing the stamp method in road tunnel safety assessment", Safety science **50**, 1806–1817 (2012).

[102] CK Allison, KM Revell, R Sears, and NA Stanton, "Systems theoretic accident model and process (stamp) safety modelling applied to an aircraft rapid decompression event", Safety science **98**, 159–166 (2017).

[103] JR Laracy, "A systems theoretic accident model applied to biodefense", Defence and Security Analysis **22**, 301–310 (2006).

[104] LV Rosa, AN Haddad, and PVR de Carvalho, "Assessing risk in sustainable construction using the functional resonance analysis method (fram)", Cognition, Technology & Work **17**, 559–573 (2015).

[105] K Lundblad, J Speziali, R Woltjer, and J Lundberg, "Fram as a risk assessment method for nuclear fuel transportation", in Proceedings of the 4th international conference working on safety, Vol. 1 (2008), pp. 223–1.

[106] R Patriarca, G Di Gravio, and F Costantino, "A monte carlo evolution of the functional resonance analysis method (fram) to assess performance variability in complex systems", Safety science **91**, 49–60 (2017).

[107] S Ahmed, HO Demirel, IY Tumer, and RB Stone, "Towards human-induced failure assessment during early design", in In tools and methods of competitive engineering (tmce 2018) (Delft University, 2018), pp. 507–520.

[108] NF Soria Zurita, RB Stone, H Onan Demirel, and IY Tumer, "Identification of human–system interaction errors during early design stages using a functional basis framework", ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering **6**, 011005 (2020).

[109] X Zhang and DB Chaffin, "Digital human modeling for computer-aided ergonomics", Handbook of Occupational Ergonomics. Taylor & Francis, CRC Press, London, Boca Raton, 1–20 (2005).

[110] VG Duffy, *Handbook of digital human modeling: research for applied ergonomics and human factors engineering* (CRC press, 2016).

[111] NI Badler, CB Phillips, and BL Webber, *Simulating humans: computer graphics animation and control* (Oxford University Press, 1993).

[112] G Colombo and U Cugini, "Virtual humans and prototypes to evaluate ergonomics and safety", Journal of Engineering Design **16**, 195–203 (2005).

[113] L Frey Law, T Xia, and A Laake, "Modeling human physical capability: joint strength and range of motion", Handbook of Digital Human Modeling, CRC Press, Boca Raton (2009).

[114] S Ahmed, J Zhang, and O Demirel, "Assessment of types of prototyping in human-centered product design", in International conference on digital human modeling and applications in health, safety, ergonomics and risk management (Springer, 2018), pp. 3–18.

[115] A Sundin and R Örtengren, "Digital human modeling for cae applications", Handbook of human factors and ergonomics, 1053–1078 (2006).

[116] L Irshad, S Ahmed, HO Demirel, and IY Tumer, "Computational functional failure analysis to identify human errors during early design stages", Journal of Computing and Information Science in Engineering **19** (2019).

[117] L Irshad, S Ahmed, O Demirel, and IY Tumer, "Identification of human errors during early design stage functional failure analysis", in Asme 2018 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers Digital Collection, 2018).

[118] L Irshad, H Onan Demirel, IY Tumer, and G Brat, "Using rio-paris flight 447 crash to assess human error and failure propagation analysis early in design", ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg **6** (2020).

[119] T Kurtoglu, IY Tumer, and DC Jensen, "A functional failure reasoning methodology for evaluation of conceptual system architectures", Research in Engineering Design **21**, 209–234 (2010).

[120] DC Jensen, "Enabling safety-informed design decision making through simulation, reasoning and analysis", PhD thesis (Oregon State University, 2012).

[121] J Hirtz, RB Stone, DA McAdams, S Szykman, and KL Wood, "A functional basis for engineering design: reconciling and evolving previous efforts", Research in engineering Design **13**, 65–82 (2002).

[122] S Sangelkar and DA McAdams, "Formalizing user activity-product function association based design rules for universal products", in Proceedings of the asme international design engineering technical conferences and computers and information in engineering conference, washington, dc (2011).

[123] S Sangelkar and DA Mcadams, "Creating actionfunction diagrams for user centric design", in American society for engineering education (American Society for Engineering Education, 2012).

[124] T Aldemir, "Computer-assisted markov failure modeling of process control systems", IEEE Transactions on reliability **36**, 133–144 (1987).

[125] N Siu, "Risk assessment for dynamic systems: an overview", Reliability Engineering & System Safety **43**, 43–73 (1994).

[126] G Cojazzi, "The dylam approach for the dynamic reliability analysis of systems", Reliability Engineering & System Safety **52**, 279–296 (1996).

[127] E Hofer, M Kloos, B Krzykacz-Hausmann, J Peschke, and M Woltereck, "An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties", Reliability Engineering & System Safety **77**, 229–238 (2002).

[128] D Harris, NA Stanton, A Marshall, MS Young, J Demagalski, and P Salmon, "Using sherpa to predict design-induced error on the flight deck", Aerospace science and technology **9**, 525–532 (2005).

[129] CE Billings, *Human-centered aircraft automation: A concept and guidelines*, Technical Memorandum (NASA Ames Research Center, 1991).

[130] NA Stanton, "Representing distributed cognition in complex systems: how a submarine returns to periscope depth", Ergonomics **57**, 403–418 (2014).

[131] J Wise, A Rio, and M Fedouach, "What really happened aboard air france 447", Popular Mechanics **6** (2011).

[132] *Final report on the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro–Paris*, Investigation Report (Bureau d'Enquêtes et d'Analyses pour la sécurité de l'aviation civile, Paris: BEA, 2012).

[133] R Garcia and L Barnes, "Multi-uav simulator utilizing x-plane", in Selected papers from the 2nd international symposium on uavs, reno, nevada, usa june 8–10, 2009 (Springer, 2009), pp. 393–406.

[134] K Ali and L Carter, "Miniature-autopilot evaluation system", J. Comput. Sci **4**, 9 (2008).

[135] A Bittar, HV Figuereido, PA Guimaraes, and AC Mendes, "Guidance software-in-the-loop simulation using x-plane and simulink for uavs", in International conference on unmanned aircraft systems (icuas) (2014), pp. 993–1002.

[136] M Ertem, "An airborne synthetic vision system with hits symbology using x-plane for a head up display", in Digital avionics systems conference, 2005. dasc 2005. the 24th, Vol. 2 (IEEE, 2005), 6–pp.

[137] L Irshad, HO Demirel, and IY Tumer, "Automated generation of fault scenarios to assess potential human errors and functional failures in early design stages", Journal of computing and information science in engineering **20** (2020).

[138] L Irshad, HO Demirel, and IY Tumer, "Using automated use case generation for early design stage functional failure and human error analysis", in Asme 2019 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers Digital Collection, 2019).

[139] L Irshad, D Hulse, HO Demirel, IY Tumer, and DC Jensen, "Quantifying the combined effects of human errors and component failures", Journal of Mechanical Design **143**, 101703 (2021).

[140] L Irshad, D Hulse, HO Demirel, IY Tumer, and DC Jensen, "Introducing likelihood of occurrence and expected cost to human error and functional failure reasoning framework", in International design engineering technical conferences and computers and information in engineering conference, Vol. 83976 (American Society of Mechanical Engineers, 2020), V008T08A031.

[141] SJ Cunning and JW Rozenblit, "Automating test generation for discrete event oriented embedded systems", Journal of Intelligent and Robotic Systems **41**, 87–112 (2005).

[142] A Junghanns, J Mauss, M Tatar, et al., "Tatar: testweaver-a tool for simulation-based test of mechatronic designs", in 6th international modelica conference, bielefeld, march 3 (Citeseer, 2008).

[143] KD Hilf, I Matheis, J Mauss, and J Rauh, "Automated simulation of scenarios to guide the development of a crosswind stabilization function", IFAC Proceedings Volumes **43**, 768–772 (2010).

[144] NA Snooke, "Automated failure effect analysis for phm of uav", Handbook of Unmanned Aerial Vehicles, 1027–1051 (2015).

[145] P Struss, "A model-based methodology for the integration of diagnosis and fault analysis during the entire life cycle", IFAC Proceedings Volumes **39**, 1157–1162 (2006).

[146] P Liggesmeyer and M Rothfelder, "Improving system reliability with automatic fault tree generation", in Digest of papers. twenty-eighth annual international symposium on fault-tolerant computing (cat. no. 98cb36224) (IEEE, 1998), pp. 90–99.

[147] H Nejad and A Mosleh, "Automated risk scenario generation using system functional and structural knowledge", in Asme 2005 international mechanical engineering congress and exposition (American Society of Mechanical Engineers, 2005), pp. 85–89.

[148] D Mercurio, L Podofillini, E Zio, and V Dang, "Identification and classification of dynamic event tree scenarios via possibilistic clustering: application to a steam generator tube rupture event", Accident Analysis & Prevention **41**, 1180–1191 (2009).

[149] IA Papazoglou, "Functional block diagrams and automated construction of event trees", Reliability Engineering & System Safety **61**, 185–214 (1998).

[150] DK Sen, JC Banks, G Maggio, and J Railsback, "Rapid development of an event tree modeling tool using cots software", in Aerospace conference, 2006 ieee (IEEE), 8–pp.

[151] C Smith, J Knudsen, K Kvarfordt, and T Wood, "Key attributes of the saphire risk and reliability analysis software for risk-informed probabilistic applications", Reliability Engineering & System Safety **93**, 1151–1164 (2008).

[152] B Rutt, U Catalyurek, A Hakobyan, K Metzroth, T Aldemir, R Denning, S Dunagan, and D Kunsman, "Distributed dynamic event tree generation for reliability and risk assessment", in Challenges of large applications in distributed environments, 2006 ieee (IEEE, 2006), pp. 61–70.

[153] MG McIntire, E Keshavarzi, IY Tumer, and C Hoyle, "Functional models with inherent behavior: towards a framework for safety analysis early in the design of complex systems", in Asme 2016 international mechanical engineering congress and exposition (American Society of Mechanical Engineers, 2016), V011T15A035–V011T15A035.

[154] N Papakonstantinou, S Sierla, B O'Halloran, and IY Tumer, "A simulation based approach to automate event tree generation for early complex system designs", in Asme 2013 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers, 2013), V02BT02A008–V02BT02A008.

[155] M Blackburn, R Busser, and A Nauman, "Why model-based test automation is different and what you should know to get started", in International conference on practical software quality and testing (2004), pp. 212–232.

[156] M Auguston, JB Michael, and MT Shing, "Environment behavior models for scenario generation and testing automation", in Acm sigsoft software engineering notes, Vol. 30, 4 (ACM, 2005), pp. 1–6.

[157] D Xu, W Xu, M Kent, L Thomas, and L Wang, "An automated test generation technique for software quality assurance", IEEE Transactions on Reliability **64**, 247–268 (2015).

[158] R Wang, LM Kristensen, H Meling, and V Stolz, "Automated test case generation for the paxos single-decree protocol using a coloured petri net model", Journal of Logical and Algebraic Methods in Programming **104**, 254–273 (2019).

[159] R Matinnejad, S Nejati, L Briand, and T Bruckmann, "Test generation and test prioritization for simulink models with dynamic behavior", IEEE Transactions on Software Engineering (2018).

[160] J Offutt and A Abdurazik, "Generating tests from uml specifications", in International conference on the unified modeling language (Springer, 1999), pp. 416–429.

[161] P Chevalley and P Thévenod-Fosse, "Automated generation of statistical test cases from uml state diagrams", in Computer software and applications conference, 2001. compsac 2001. 25th annual international (IEEE, 2001), pp. 205–214.

[162] V Santiago, ASM Do Amaral, NL Vijaykumar, MdF Mattiello-Francisco, E Martins, and OC Lopes, "A practical approach for automated test case generation using statecharts", in 30th annual international computer software and applications conference (compsac'06), Vol. 2 (IEEE, 2006), pp. 183–188.

[163] S Pradhan, M Ray, and SK Swain, "Transition coverage based test case generation from state chart diagram", Journal of King Saud University-Computer and Information Sciences (2019).

[164] R Verma and R Bhatia, "Behavior based automated test case generation for object oriented systems", International Journal of Computer Applications **54** (2012).

[165] SK Swain, DP Mohapatra, and R Mall, "Test case generation based on state and activity models.", Journal of Object Technology **9**, 1–27 (2010).

[166] P Sapna and H Mohanty, "Automated scenario generation based on uml activity diagrams", in 2008 international conference on information technology (IEEE, 2008), pp. 209–214.

[167] A Shanthi, G MohanKumar, et al., "A novel approach for automated test path generation using tabu search algorithm", International Journal of Computer Applications **48**, 28–34 (2012).

[168] H Stallbaum, A Metzger, and K Pohl, "An automated technique for risk-based test case generation and prioritization", in Proceedings of the 3rd international workshop on automation of software test (ACM, 2008), pp. 67–70.

[169] FAD Teixeira and GB e Silva, "Easytest: an approach for automatic test cases generation from uml activity diagrams", in *Information technology-new generations* (Springer, 2018), pp. 411–417.

[170] C Nebut, F Fleurey, Y Le Traon, and JM Jezequel, "Automatic test generation: a use case driven approach", IEEE Transactions on Software Engineering **32**, 140–155 (2006).

[171] M Sarma and R Mall, "Automatic test case generation from uml models", in 10th international conference on information technology (icit 2007) (IEEE, 2007), pp. 196–201.

[172] N Raza, A Nadeem, and MZZ Iqbal, "An automated approach to system testing based on scenarios and operations contracts", in Seventh international conference on quality software (qsic 2007) (IEEE, 2007), pp. 256–261.

[173] M Prasanna and K Chandran, "Automatic test case generation for uml object diagrams using genetic algorithm", Int. J. Advance. Soft Comput. Appl **1**, 19–32 (2009).

[174] T Aven, "Risk assessment and risk management: review of recent advances on their foundation", European Journal of Operational Research **253**, 1–13 (2016).

[175] M Stamatelatos, W Vesely, J Dugan, J Fragola, J Minarick, and J Railsback, *Fault tree handbook with aerospace applications*, tech. rep. (NASA, 2002).

[176] KC Kapur and M Pecht, *Reliability engineering* (Wiley, Hoboken, New Jersey, 2014).

[177] BM O'Halloran, C Hoyle, IY Tumer, and RB Stone, "The early design reliability prediction method", Research in Engineering Design **30**, 489–508 (2019).

[178] SJ Rhee and K Ishii, "Using cost based fmea to enhance reliability and serviceability", Advanced Engineering Informatics **17**, 179–188 (2003).

[179] S Kmenta and K Ishii, "Scenario-based failure modes and effects analysis using expected cost", J. Mech. Des. **126**, 1027–1035 (2004).

[180] A von Ahsen, "Cost-oriented failure mode and effects analysis", International Journal of Quality & Reliability Management **25**, 466–476 (2008).

[181] N Yodo and P Wang, "Engineering resilience quantification and system design implications: a literature survey", Journal of Mechanical Design **138**, 111408 (2016).

[182] E Miller-Hooks, X Zhang, and R Faturechi, "Measuring and maximizing resilience of freight transportation networks", Computers & Operations Research **39**, 1633–1643 (2012).

[183] CA MacKenzie and C Hu, "Decision making under uncertainty for design of resilient engineered systems", Reliability Engineering & System Safety **192**, 106171 (2019).

[184] D Hulse, C Hoyle, K Goebel, and IY Tumer, "Optimizing function-based fault propagation model resilience using expected cost scoring", in Asme 2018 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers Digital Collection, 2018).

[185] UD of Transportation, "Revised departmental guidance 2013: treatment of the value of preventing fatalities and injuries in preparing economic analyses", (2013).

[186] AV Aho and JE Hopcroft, *The design and analysis of computer algorithms* (Pearson Education India, 1974).

[187] P Giudici, GH Givens, and BK Mallick, *Wiley series in computational statistics* (Wiley Online Library, 2013).

[188] ' Quanterion Solutions Incorporated, *Nonelectronic parts reliability data 2016* (Reliability Analysis Center, 2016).

[189] ' Quanterion Solutions Incorporated, *Electronic parts reliability data 2014* (Reliability Analysis Center, 2014).

[190] ' Quanterion Solutions Incorporated, *Failure mode/mechanism distributions* (Reliability Analysis Center, 2016).

[191] BM O'Halloran, RB Stone, and IY Tumer, "A failure modes and mechanisms naming taxonomy", in 2012 proceedings annual reliability and maintainability symposium (IEEE, 2012), pp. 1–6.

[192] W Denson, G Chandler, W Crowell, A Clark, and P Jaworski, *Nonelectronic parts reliability data - 1995*, tech. rep. NPRD-95 (Reliability Analysis Center, Rome, NY, 1994).

[193] W Crowell, W Denson, P Jaworski, and D Mahar, *Failure mode/mechanism distributions 1997*, tech. rep. FMD-97 (Reliability Analysis Center, Rome, NY, 1997).

[194] IM Sobol, "Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates", Mathematics and computers in simulation **55**, 271–280 (2001).

[195] A Saltelli, "Making best use of model evaluations to compute sensitivity indices", Computer physics communications **145**, 280–297 (2002).

[196] T Aven and R Flage, "Use of decision criteria based on expected values to support decision-making in a production assurance and safety setting", Reliability Engineering & System Safety **94**, 1491–1498 (2009).

[197] MK Wright, L Stokes, and JS Dyer, "Reliability and coherence of causal, diagnostic, and joint subjective probabilities", Decision Sciences **25**, 691–709 (1994).

[198] D Hulse, C Hoyle, IY Tumer, and K Goebel, "Decomposing incentives for early resilient design: method and validation", in Asme 2019 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers).

[199] V Kattakuri and JH Panchal, "Spacecraft failure analysis from the perspective of design decision-making", in Asme 2019 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers, 2019).

[200] ME Kreye, YM Goh, and LB Newnes, "Manifestation of uncertainty-a classification", in Ds 68-6: proceedings of the 18th international conference on engineering design (iced 11), impacting society through engineering design, vol. 6: design information and knowledge, lyngby/copenhagen, denmark, 15.-19.08. 2011 (2011).

[201] L Irshad, HO Demirel, and IY Tumer, "Human error and functional failure reasoning framework: how does it scale?", in International design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers, 2021).

[202] CY Baldwin, KB Clark, KB Clark, et al., *Design rules: the power of modularity*, Vol. 1 (MIT press, 2000).

[203] JK Gershenson, G Prasad, and Y Zhang, "Product modularity: definitions and benefits", Journal of Engineering design **14**, 295–313 (2003).

[204] CY Baldwin and KB Clark, "Modularity in the design of complex engineering systems", in *Complex engineered systems* (Springer, 2006), pp. 175–205.

[205] KM Holtta and MP Salonen, "Comparing three different modularity methods", in Asme 2003 international design engineering technical conferences and computers and information in engineering conference (American Society of Mechanical Engineers Digital Collection, 2003), pp. 533–541.

[206] HS Walsh, A Dong, and IY Tumer, "An analysis of modularity as a design rule using network theory", Journal of Mechanical Design **141** (2019).

[207] SK Ethiraj and D Levinthal, "Modularity and innovation in complex systems", Management science **50**, 159–173 (2004).

[208] K Hölttä-Otto and O De Weck, "Degree of modularity in engineering systems and products with technical and business constraints", Concurrent Engineering **15**, 113–126 (2007).

[209] S Eum, S Arakawa, and M Murata, "Toward bio-inspired network robustness-step 1. modularity", in 2007 2nd bio-inspired models of network, information and computing systems (IEEE, 2007), pp. 84–87.

[210] TD Tran and YK Kwon, "The relationship between modularity and robustness in signalling networks", Journal of The Royal Society Interface **10**, 20130771 (2013).

[211] SD Eppinger and TR Browning, *Design structure matrix methods and applications* (MIT press, 2012).

[212] A Ericsson and G Erixon, *Controlling design variants: modular product platforms* (Society of Manufacturing Engineers, 1999).

[213] Y Li, Z Wang, L Zhang, X Chu, and D Xue, "Function module partition for complex products and systems based on weighted and directed complex networks", Journal of Mechanical Design **139** (2017).

[214] B Nepal, L Monplaisir, and N Singh, "Integrated fuzzy logic-based model for product modularization during concept development phase", International Journal of Production Economics **96**, 157–174 (2005).

[215] F Borjesson and K Hölttä-Otto, "A module generation algorithm for product architecture based on component interactions and strategic drivers", Research in Engineering Design **25**, 31–51 (2014).

[216] Q Cheng, Y Guo, Z Liu, G Zhang, and P Gu, "A new modularization method of heavy-duty machine tool for green remanufacturing", Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science **232**, 4237–4254 (2018).

[217] RB Stone, KL Wood, and RH Crawford, "A heuristic method for identifying modules for product architectures", Design studies **21**, 5–31 (2000).

[218] N Chiriac, K Hölttä-Otto, D Lysy, and E Suk Suh, "Level of modularity and different levels of system granularity", Journal of Mechanical Design **133** (2011).

[219] JM Shultz, MP Garcia-Vera, CG Santos, J Sanz, G Bibel, C Schulman, G Bahouth, Y Dias Guichot, Z Espinel, and A Rechkemmer, "Disaster complexity and the santiago de compostela train derailment", Disaster health **3**, 11–31 (2016).

[220] ' National Transportation Safety Board, *Conrail Freight Train Derailment with Vinyl Chloride Release*, Accident Report NTSB/RAR-14/01 (National Transportation Safety Board, Washington DC, 2012).

[221] S Ahmed, L Irshad, MS Gawand, and HO Demirel, "Integrating human factors early in the design process using digital human modelling and surrogate modelling", Journal of Engineering Design **32**, 165–186 (2021).

[222] VG Duffy, "Modified virtual build methodology for computer-aided ergonomics and safety", Human Factors and Ergonomics in Manufacturing & Service Industries **17**, 413–422 (2007).

[223] *Jack fact sheet*, `https://www.plm.automation.siemens.com/media/store/en_us/4917_tcm1023-4952_tcm29-1992.pdf`, (Accessed on 07/17/2021).

[224] M Gawand, "Automating digital human modeling for task simulation and ergonomic evaluation to consider emergency ergonomics early in design", (2019).

[225] J Gertler, "F-35 joint strike fighter (jsf) program", in (LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE, 2012).

[226] A Mikes, K Edmonds, RB Stone, and B DuPont, "Autofunc: a python package for automating and verifying functional modeling", Journal of Open Source Software **6**, 2362 (2021).

APPENDICES

Appendix A: HEART Generic Tasks and Error Producing Conditions

Table 1: HEART GENERIC TASK TYPES [25]

Generic Task Type	**Description**	**Proposed Nominal Human Unreliability**
(A)	Totally unfamiliar, performed at speed with no real idea of likely consequences	0.55
(B)	Shift or restore system to a new or original state on a single attempt without supervision or procedures	0.26
(C)	Complex task requiring high level of comprehension and skill	0.16
(D)	Fairly simple task performed rapidly or given scant attention	0.09
(E)	Routine, highly-practiced, rapid task involving relatively low level of skill	0.02
(F)	Restore or shift a system to original or new state following procedures, with some checking	0.003
(G)	Completely familiar, well-designed, highly-practised, routine task occurring several times per hour, performed to highest possible standards by highly-motivated, highly-trained and experienced person, totally aware of implications of failure, with time to correct potential error, but without the benefit of significant job aids	0.004
(H)	Respond correctly to system command even when there is an augmented or automated supervisory system providing accurate interpretation of system state	0.00002
(M)	Miscellaneous task for which no description can be found	0.03

Table 2: HEART ERROR PRODUCING CONDITION FACTORS [25]

EPC#	**Error Producing Conditions**	**Maximum Effect Factor**
1	Unfamiliarity with a situation which is potentially important but which only occurs infrequently or which is novel	17
2	A shortage of time available for error detection and corrections	11
3	A low signal to noise ratio	10

4	A means of suppressing or over-riding information or features which is too easily accessible	9
5	No means of conveying spatial and functional information to operators in a form which they can readily assimilate	8
6	A mismatch between an operator's model of the world and that imagined by a designer	8
7	No obvious means of reversing an unintended action	8
8	A channel capacity overload, particularly one caused by simultaneous presentation of non-redundant information	6
9	A need to unlearn a technique and apply one which requires the application of an opposing philosophy	6
10	The need to transfer specific knowledge from task to task without loss	5.5
11	Ambiguity in the required performance standards	5
12	A mismatch between perceived and real risk	4
13	Poor, ambiguous or ill-matched system feedback	4
14	No clear direct and timely confirmation of an intended action from the portion of the system over which control is to be exerted	4
15	Operator inexperience (e.g. a newly-qualified tradesman, but not an "expert")	3
16	An impoverished quality of information conveyed by procedures and person/person interaction	3
17	Little or no independent checking or testing of output	3
18	A conflict between immediate and long-term objectives	2.5
19	No diversity of information input for veracity checks	2.5
20	A mismatch between the educational achievement level of an individual and the requirements of the task	2
21	An incentive to use other more dangerous procedures	2

22	Little opportunity to exercise mind and body outside the immediate confines of a job	1.8
23	Unreliable instrumentation (enough that it is noticed)	1.6
24	A need for absolute judgments which are beyond the capabilities or experience of an operator	1.6
25	Unclear allocation of function and responsibility	1.6
26	No obvious way to keep track of progress during an activity	1.4
27	A danger that finite physical capabilities will be exceeded	1.4
28	Little or no intrinsic meaning in a task	1.4
29	High-level emotional stress	1.3
30	Evidence of ill-health amongst operatives, especially fever	1.2
31	Low workforce morale	1.2
32	Inconsistency of meaning of displays and procedures	1.2
33	A poor or hostile environment (below 75% of health or life-threatening severity)	1.1
34	Prolonged inactivity or highly repetitious cycling of low mental workload tasks	1.1 (for 1st half-hour)/1.0 (for each hour thereafter)
35	Disruption of normal work-sleep cycles	1.1
36	Task pacing caused by the intervention of others	1.06
37	Additional team members over and above those necessary to perform task normally and satisfactorily	1.03 per additional team member
38	Age of personnel performing perpetual tasks	1.02

Appendix B: Simulation Inputs for the Risk Quantification Model in Chapter 4

Table 3: COMPONENT FAILURE RATES, DISTRIBUTIONS, REPAIR COST, AND RECOVERY TIMES

Component	**Failure Modes**	**Failure Rate (NPRD-95) [192] per million hours**	**Distributions (FMD-97)[193]**	**Notes**	**Repair Cost ($)**[1]	**Repair Time (h)**[1]
Tank	Leak	1.616	100%	Summary Data from Storage Tank was used due to its similarities with the coolant tank case study	20000	24
Valve	StuckOpen	3.0764	47.44%	A hydraulic valve was chosen. The failure mode leak was omitted because it is modeled in the pipe and modeling it here will be redundant	10000	6
	StuckClose		52.56%		10000	6
Pipe	Leak	0.4734	7.42%	Summary data of the component Piping was chosen for the pipe failure rate.	10000	6
	Ruptured		92.58%	The Failure Mode Broken is Considered as ruptured	15000	12

Table 4: HUMAN ACTIONS, DESIGNATIONS, AND RELATED GENERIC TASKS

Actions	**Designation**	**Generic Task**
Look	Both	E- 0.02
Detect	Diagnosis	M - 0.03
Reach	Action	D - 0.09
Grasp	Action	D - 0.09
Turn	Action	F - 0.003

Table 5: COSTS OF FUNCTION FAILURES

	Performance Cost[1]		Immediate Cost[1]
Function	Degraded	Lost	Lost
Import Liquid	35000	175000	0
Guide Liquid	60000	300000	0
Transfer Liquid	80000	400000	2000000
Store Liquid	100000	500000	5000000
Supply Liquid	40000	200000	0
Transfer Liquid	25000	125000	3000000
Guide Liquid	75000	375000	0
Export Liquid	30000	150000	0

[1]Details on how these values were assigned are provided in section 4.5.

Table 6: HUMAN ERROR PRODUCING CONDITIONS AND THEIR PROPORTION OF EFFECTS FOR EACH ACTION

Error Producing Conditions (EPC)		EPC Propotion of Effects[1]									
		Look		Detect		Reach		Grasp		Turn	
Inlet Valve	OutletValve	Inlet	Outlet	Inlet	Outlet	Inlet	Outlet	Inlet	Outlet	Inlet	Outlet
EPC2-11	EPC2-11	0	0	0.1	0.2	0.1	0.1	0	0	0.4	0.6
EPC10-5.5	EPC10-5.5	0	0	0.2	0.2	0	0	0	0	0.2	0.2
EPC13-4	EPC13-4	0.1	0.2	0	0	0.1	0.1	0	0	0	0
EPC14 - 4	EPC14 - 4	0.6	0.3	0.1	0.1	0	0	0	0	0	0
EPC17-3	EPC17-3	0	0	0	0	0	0	0	0	0.6	0.4
EPC34 - 1.1	EPC34 - 1.1	0.9	0.9	0.6	0.6	0	0	0	0	0	0