INL/EXT-15-36051
Revision 0

# Light Water Reactor Sustainability Program

## A Research Roadmap for Computation-Based Human Reliability Analysis

**July 2015**

DOE Office of Nuclear Energy

INL/EXT-15-36051
Revision 0

# Light Water Reactor Sustainability Program

# A Research Roadmap for Computation-Based Human Reliability Analysis

**Ronald Boring, Diego Mandelli, Jeffrey Joe, Curtis Smith**
***Idaho National Laboratory***

**Katrina Groth**
***Sandia National Laboratories***

**July 2015**

**Idaho National Laboratory**
**Idaho Falls, Idaho 83415**

**http://www.inl.gov/lwrs**

**Prepared for the**
**U.S. Department of Energy**
**Office of Nuclear Energy**
**Under DOE Idaho Operations Office**
**Contract DE-AC07-05ID14517**

# ABSTRACT

The United States (U.S.) Department of Energy (DOE) is sponsoring research through the Light Water Reactor Sustainability (LWRS) program to extend the life of the currently operating fleet of commercial nuclear power plants. The Risk Informed Safety Margin Characterization (RISMC) research pathway within LWRS looks at ways to maintain and enhance the safety margins of these plants. The RISMC pathway includes significant developments in the area of thermohydraulics code modeling and the development of tools to facilitate dynamic probabilistic risk assessment (PRA). PRA is primarily concerned with the risk of hardware systems at the plant; yet, hardware reliability is often secondary in overall risk significance to human errors that can trigger or compound undesirable events at the plant. This report highlights ongoing efforts to develop a computation-based approach to human reliability analysis (HRA). This computation-based approach differs from existing static and dynamic HRA approaches in that it: (i) interfaces with a dynamic computation engine that includes a full-scope plant model, and (ii) interfaces with a PRA software toolset. The computation-based HRA approach presented in this report is called the Human Unimodel for Nuclear Technology to Enhance Reliability (HUNTER). HUNTER incorporates in a hybrid fashion elements of existing HRA methods to interface with new computational tools developed under the RISMC pathway. The goal of this research effort is to account for human performance more accurately than existing approaches, thereby minimizing modeling uncertainty found in current plant risk models.

This page intentionally left blank

# CONTENTS

# FIGURES

# TABLES

# ACRONYMS

| | |
|---|---|
| ADAPT | Adaptive Dynamic Accident Progression Trees |
| ADS | Accident Dynamic Simulator |
| ATHEANA | A Technique for Human Error Analysis |
| BN | Bayesian Network |
| BWR | Boiling Water Reactor |
| CAPS | Crew Activity Primitives |
| CARA | Controller Action Reliability Assessment |
| CDF | Cumulative Distribution Function |
| CES | Cognitive Environment Simulation |
| CREATE | Cognitive Reliability Assessment Technique |
| DET | Dynamic-Event Trees |
| DDET | Dynamic Decision Event Tree |
| DOE | Department Of Energy |
| DURESS | Dual Reservoir System Simulator |
| DYLAM | Dynamic Logical Analytical Methodology |
| GRS | Gesellschaft für Anlagen- und Reaktorsicherheit |
| HEP | Human Error Probability |
| HERA | Human Event Repository and Analysis |
| HFE | Human Failure Event |
| HPM | Human Performance Modeling |
| HRA | Human Reliability Analysis |
| HSI | Human-System Interface |
| HUNTER | Human Unimodel for Nuclear Technology to Enhance Reliability |
| I & C | Instrumentation and Control |
| IAEA | International Atomic Energy Agency |
| IDAC | Information Decision and Action in Crew Context |
| IDHEAS | Integrated Decision-Tree Human Event Analysis System |
| INL | Idaho National Laboratory |
| LHS | Latin Hypercube Sampling |
| LWRS | Light Water Reactor Sustainability |
| MC | Monte-Carlo |
| MCDET | Monte-Carlo and Dynamic-Event Trees |
| MELCOR | Methods for Estimation of Leakages and Consequences of Releases |
| MOOSE | Multiphysics Object Oriented Simulation Environment |
| NARA | Nuclear Action Reliability Assessment |
| NASA | National Aeronautics and Space Administration |
| NPP | Nuclear Power Plants |
| NUCLARR | Nuclear Computerized Library for Assessing Reactor Reliability |
| NUREG | US Nuclear Regulatory Commission Regulation |
| OSU | The Ohio State University |
| PCA | Plant Control Actions |
| PIE | Plant Interface Element |
| PORV | Power-Operated Relief Valve |
| PRA | Probabilistic Risk Assessment |

| | |
|---|---|
| PSF | Performance Shaping Factor |
| PDF | Probabilistic Distribution Function |
| R & D | Research & Development |
| RAVEN | Reactor Analysis and Virtual control ENviroment |
| RELAP | Reactor Excursion and Leak Analysis Program |
| RISMC | Risk Informed Safety Margin Characterization |
| RO | Reactor Operator |
| ROMs | Reduced Order Models |
| RVACS | Reactor Vessel Auxiliary Cooling System |
| SACADA | Scenario Authoring, Characterization, and Debriefing Application |
| SBO | Station Blackout |
| SFPC | Spent Fuel Pool Cooling |
| SG | Steam Generator |
| SPAR-H | Standardized Plant Analysis Risk-Human Reliability Analysis |
| SRO | Senior Reactor Operator |
| STA | Shift Technical Advisor |
| THERP | Technique for Human Error Rate Prediction |
| UCLA | University of California, Los Angeles |
| UI | University of Idaho |
| UMD | University of Maryland |
| U.S. | United States |

# ACKNOWLEDGEMENTS

This report was made possible through funding by the U.S. Department of Energy (DOE) Light Water Reactor Sustainability (LWRS) Program. We would like to thank Heather Medema of INL, Tina Miyake of Idaho State University, and Rachel Benish Shirley of the Ohio State University, who contributed their talents to making this project successful. Finally, we are indebted to the many pioneers in the field of dynamic human reliability analysis, without whom this project would not be possible.

This page intentionally left blank

# A Research Roadmap for Computation-Based Human Reliability Analysis

## 1. INTRODUCTION

### 1.1 Introducing Computation-Based Human Reliability Analysis

Probabilistic risk assessment (PRA) assesses plant safety through quantitative risk measures. Typically measured as conditional core damage frequency or probability, the output of the PRA accounts for the likelihood of damage to the plant fuel, containment, or surrounding environment in the event of failures to specific hardware systems. Hardware systems are operated by humans; as such, human actions or inactions are integral to the overall analysis of risk.

Mosleh (2014) and Coyne and Siu (2013) have emphasized the importance of computational approaches to PRA. These approaches, which use dynamic simulations of events at plants, potentially provide greater accuracy in overall risk modeling. In this report, we introduce the concept of *computation-based* human reliability analysis (HRA). The key elements of this research approach are:

- Use of computational techniques, namely simulation and modeling, to integrate virtual operator models with virtual plant models.
- Dynamic modeling of human cognition and actions.
- Incorporation of these respective elements into a PRA framework.

The goal of this research is to achieve a high fidelity causal representation of the role of the human operator at the plant. By thoroughly accounting for human actions, the uncertainty surrounding PRA can be reduced. Additionally, by modeling human actions dynamically, it is possible to model types of activities and events in which the human role is not clearly understood or predicted, e.g., unexampled events such as severe accidents. The ability to simulate the role of the human complements and, indeed, greatly enhances other PRA modeling efforts currently underway.

### 1.2 Overview of Risk Informed Safety Margin Characterization

One area of the U.S. Department of Energy's (DOE's) Light Water Reactor Sustainability (LWRS) project is the Risk Informed Safety Margin Characterization (RISMC) pathway (Smith, Rabiti & Martineau, 2011). RISMC research centers on understanding not just the frequency of an event like core damage, but also how close the plant is (or is not) to key safety-related events and how the plant might increase the *safety margin*. A *safety margin* can be characterized in one of two ways:

- A deterministic margin, typically defined by the ratio (or, alternatively, the difference) of a capacity (i.e., strength) over the load.

- A probabilistic margin, defined by the probability that the load exceeds the capacity. A probabilistic safety margin is a numerical value quantifying the probability that a safety metric (e.g., for an important observable process such as clad temperature) will be exceeded under accident scenario conditions.

The RISMC Pathway uses the probabilistic methods to determine safety margins and quantify their impacts on reliability and safety. As part of quantification, both probabilistic (via risk simulation) and mechanistic (via system simulators) approaches are used, as represented in Figure 1. Probabilistic analysis is represented by the risk analysis while mechanistic analysis is represented by the plant physics calculations. In the plant simulation, all deterministic aspects that characterize system dynamics (e.g., thermo-hydraulic, thermo-mechanics, neutronics) are coupled to one another.

The risk simulation (see Figure 1) contains all deterministic elements that impact accident evolution from a controller point of view such as:

- Safety systems control logic, and
- Accident scenario initial and boundary conditions.

Additionally, there are a number of stochastic elements introduced, including:

- System/components failures, and
- Stochastic perturbation of internal elements within the physics simulation.

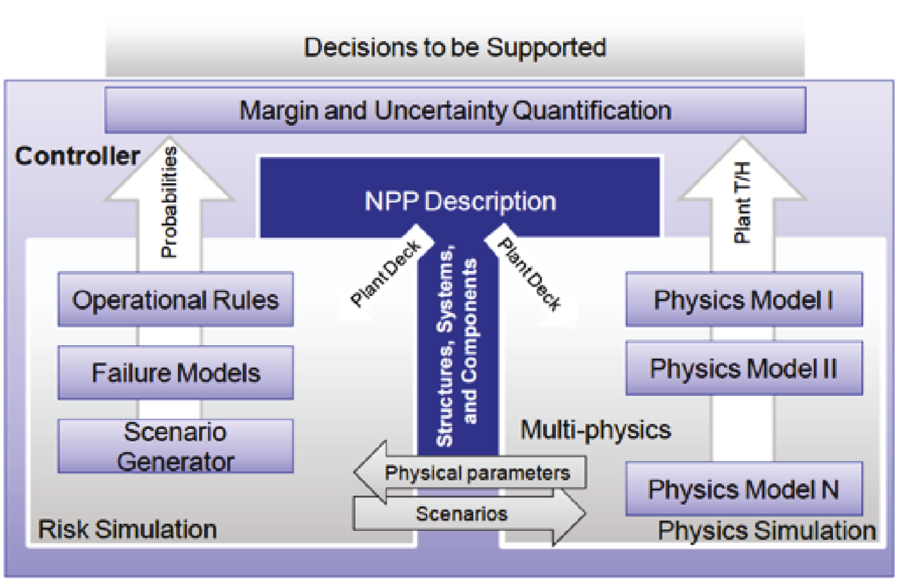


Figure 1: The probabilistic and mechanistic approaches used to support RISMC analysis

Accident evolution is also influenced by the response of reactor crews, operators, and staff. These interactions can be classified in two ways:

- *Deterministic:* Through accident management procedures where the set of *actions* to be taken is followed in a sequential fashion.
- *Stochastic:* Time to perform an action is not immediate but aleatory; in addition, a wrong action can be performed (i.e., error of commission) or such action can be neglected (i.e., error of omission).

In past RISMC research studies, human interactions were loosely considered (Mandelli et al., 2013). However, throughout the history of nuclear industry (INSAG, 1992; NRC, 2014; INPO, 2011), human interaction has played a primary role in accident evolution.

In Boring et al. (2014), researchers performed a literature overview of methods that aim to model and quantify the impact of human interactions on plant safety. Both static and simulation based human reliability approaches were evaluated. Further, simulation-based human reliability models were analyzed and deemed the most suited to fit into the RISMC approach due to the intrinsic coupling between human interactions and accident evolution. This report introduces the means by which human reliability modeling can be performed within the RISMC approach.

As mentioned earlier, the RISMC approach heavily relies on multi-physics system simulator codes such as the Reactor Excursion and Leak Analysis Program Version 7 (RELAP-7) code (David et al., 2012) coupled with stochastic analysis tools such as the Risk Analysis and Virtual control ENviroment (RAVEN) tool (Alfonsi et al., 2013). From a mathematical point of view, a single simulator run can be represented as a single trajectory in the phase space.[a] The evolution of such a trajectory in the phase space is modeled as follows:

$$\frac{\partial \boldsymbol{\theta}(t)}{\partial t} = \boldsymbol{\mathcal{H}}(\boldsymbol{\theta}, \boldsymbol{s}, t) \tag{1}$$

where:

- $\boldsymbol{\theta} = \boldsymbol{\theta}(t)$ represents the temporal evolution of a simulated accident scenario, i.e., $\boldsymbol{\theta}(t)$ represents the parameter space of a single simulation run
- $\boldsymbol{\mathcal{H}}$ is the simulator code that describes how $\boldsymbol{\theta}$ evolves in time
- $\boldsymbol{s} = \boldsymbol{s}(t)$ represents the status of components and systems of the simulator (e.g., status of emergency core cooling system)
- $t$ is time in scenario space (vs. simulation time)

Using the RISMC approach, the PRA analysis is performed by:

1. Associating a probability distribution function (pdf) with the set of stochastic system parameters $\boldsymbol{s}$ (e.g., timing of events).
2. Performing stochastic sampling of the pdfs defined in Step 1.
3. Performing a simulation run given $\boldsymbol{s}$ sampled in Step 2, i.e., solve Eq. (1).
4. Repeating Steps 2 and 3 *M* times.

[a] *Phase space* is the space composed by all degrees of freedom within a system.

5. Transform the results into user-defined decision metrics or figures of merit, such as the Core Damage Probability ($P_{CD}$).

Human interactions contribute to the dynamics of accident evolution, because humans can change the status of components and systems. Thus, they can be modeled within the set of variable $\boldsymbol{s}(t)$. Within the RISMC approach, the set of human interactions have a pdf associated with them, representing the uncertainty in carrying out specific actions or functions.

### 1.2.1 Limit Surface

Research carried out under the RISMC project is not limited to reporting results regarding the probability of core damage and containment breach events. Rather, the RISMC approach aims to explore the vulnerabilities and limitations of the system under consideration by analyzing the space of possible events. As an example, the RISMC approach aims to evaluate a set of limit surfaces (Mandelli & Smith, 2012). A limit surface represents the boundaries in the input space (i.e., $d$-dimensional space; each dimension is one the $d$ sampled variables) that separate failure region (i.e., characterized by the undesired simulation outcome; e.g., core damage) from success region (i.e., characterized by the desired simulation outcome; e.g., max clad temperature below 2200° F). Figure 2 provides an example of the limit surface evaluation for a station blackout (SBO) occurring at a boiling water reactor (BWR) power station.

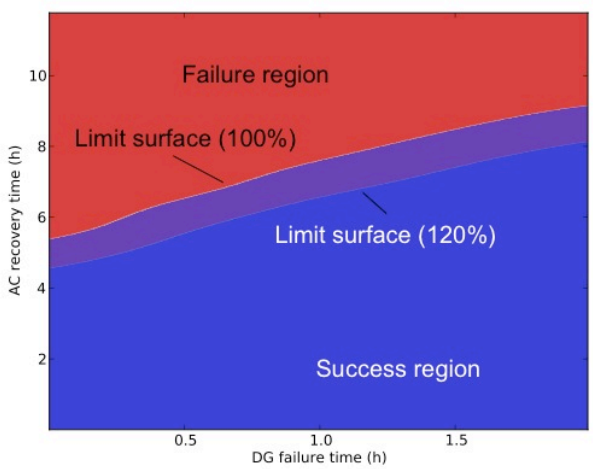


Figure 2: Example of the limits surface for a BWR SBO test case (Helton & Davis, 2003)

Per se, the limit surface has pure deterministic information; the stochastic information is generated when the probability of an undesired event occurring (e.g., core damage) $P_{CD}$ is determined as:

$$P_{CD} = \int_{\substack{failure \\ region}} pdf(\varpi)\, d\varpi \tag{2}$$

Equation 2 indicates that $P_{CD}$ is equal to the area of the failure region weighted by the probability of being in the failure region itself (through the probability distribution function, $pdf(\varpi)$). As noted, Figure 2 shows the limit surface in a 2-dimensional space generated in Mandelli et al. (2013) using RAVEN coupled with RELAP-7 for a BWR SBO initiating event. As part of the analysis, focus was directed towards the evaluation of the safety impact of power uprate (i.e., reactor core power increased from 100 to 120%). Researchers evaluated both the increased core damage probability, $\Delta P_{CD}$, and the limit surface for both 100 and 120% reactor core power level. Note that $\Delta P_{CD}$ can be written as the same integral indicated in Eq. 2 but evaluated only in the expanded failure region ($\Delta\Omega_{Failure}$):

$$\Delta P_{CD} = \int_{\Delta\Omega_{Failure}} pdf(\varpi)\, d\varpi \qquad (3)$$

## 1.3 First-Year RISMC Research on Human Reliability

In fiscal year 2015, the RISMC pathway is undertaking three phases of research directed at clarifying the role of human reliability in support of overall risk modeling. These phases are summarized in Figure 3. In the first phase, captured in Boring et al. (2014), we compared approaches to modeling human reliability with a particular emphasis on cognitive modeling architectures that could support simulation. The present report summarizes the second phase of research, which seeks to articulate a framework for computation-based HRA. This framework attempts to move HRA beyond its static origins to use computational tools now available to model dynamic aspects of operator performance. A final, forthcoming phase of this year's research will aim to demonstrate the framework in practice with a simulation case study. The case study will build on a flooding example, but the concepts will be more broadly applicable to other human reliability contexts.

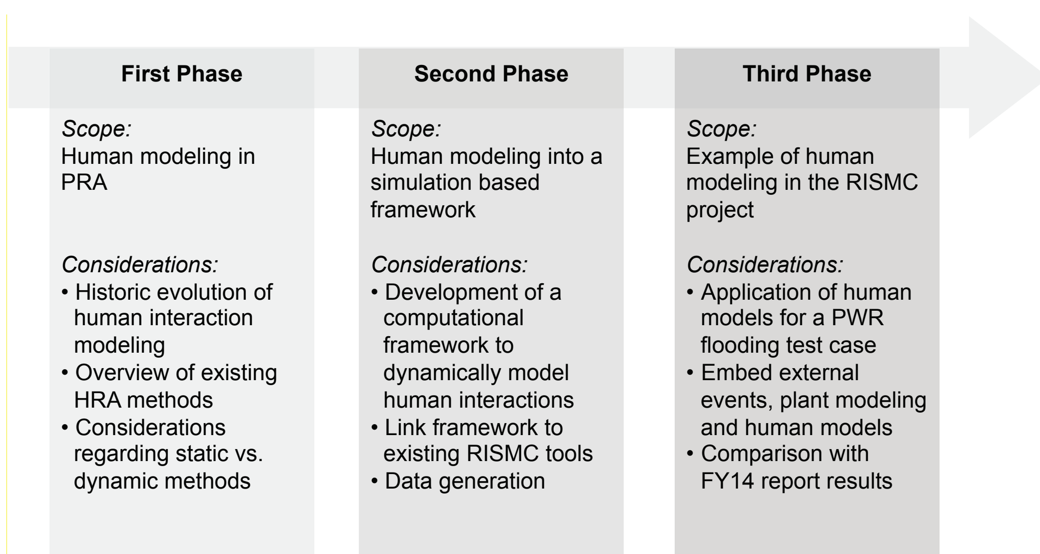


Figure 3: First-year human reliability modeling under RISMC

## 1.4 Overview of Report

This report is divided into six chapters that explain the concept and components of computation-based HRA as well as the research needs to implement this framework. The report chapters cover:

- *Chapter 1:* The current chapter, which overviews RISMC and the link to HRA.
- *Chapter 2:* This chapter presents the selected paradigms of HRA, detailing key distinctions between static, dynamic, and computation-based HRA.
- *Chapter 3:* This chapter introduces the concept of the computational engine that integrates the virtual plant and operator models.
- *Chapter 4:* This chapter introduces a research framework for dynamic elements of HRA and discusses how models of operator performance can be developed to interface with the computational engine.
- *Chapter 5:* This chapter provides background on key research areas required in implementing computation-based HRA.
- *Chapter 6:* The final chapter summarizes conclusions and discusses the path forward for the next phase of RISMC research on computation-based HRA.

# 2. SELECTED PARADIGMS OF HUMAN RELIABILITY ANALYSIS

## 2.1 Introduction

In this chapter, we discuss three different paradigms of HRA—static, dynamic, and computation-based. While the distinctions between static and dynamic HRA are well documented (e.g., Boring, 2007; Ekanem & Mosleh, 2014), computation-based HRA is not common nomenclature. As such, this chapter provides definitional background to the three different types of HRA and ultimately makes the argument for the importance of computation-based HRA.

## 2.2 The Current Paradigm: Static HRA

Static HRA supports PRA by considering the human contribution to overall system risk. HRA may be successfully integrated into PRA in a well-established process (Bell & Swain, 1983; EPRI, 1992; IEEE, 1997). The key to this integration is the human failure event (HFE), which represents a clustering of human activities related to the operation of a particular system or component. The HFE can be quantified using any of a number of HRA methods (for recent surveys, see Bell & Holroyd, 2009; Chandler et al., 2006; and Kolaczkowski et al., 2005). The HFE is integrated into the event trees used in the PRA. Often the clustering of activities under the HFE is done using fault tree logic. In practice, the HFE is defined as the entirety of human actions related to the human interaction with a particular system. In other words, the HFE is defined top-down, from the PRA level of interest, to encompass all human actions that can contribute to the fault of a component or system modeled in the PRA.

Static HRA mimics the predominance of static PRA. The key point in static HRA and PRA is that events are analyzed for an assumed, e.g., typical, window of time. The HFE for static HRA does not change as a function of time or the event progression; the event sequences are fixed in the HRA, and the analysis represents a snapshot of time. Either the analysis represents a very generic context in which the event would occur, or the analysis is agnostic to time, meaning that time evolution is simply not factored into the calculation of the human error probability (HEP). Other performance shaping factors (PSFs) apart from time drive the quantification of the HEP.

As Boring, Joe, and Mandelli (2015) and Joe et al. (2015) point out, widely used HRA methods, such as the Standardized Plant Analysis Risk-Human Reliability Analysis (SPAR-H) method (Gertman et al., 2005), are static. They do not provide a dynamic account of human actions or how the PSFs can dynamically modify the HEP over time. SPAR-H and similar methods generally entail three steps:

- Identification of human failure events (often through task analysis),
- Assessment of context (e.g., via assigning states to PSFs and other contextual factors), and
- Computation of an HEP (generally via a linear equation defining how the state of the contextual variables changes a nominal HEP for the task and/or HFE).

A human reliability analyst using SPAR-H would first identify HFEs involving risk significant human errors and successful human actions. The analyst would then use SPAR-H to model and quantify the operator diagnoses (e.g., cognitive activities) and operator actions (e.g., behaviors) associated with the identified HFEs, starting with nominal HEPs, and then multiplying the nominal HEPs by any or all of the eight PSF modifiers provided in the method.

SPAR-H calculates an HEP based on a static rating of PSFs. In essence:

$$HEP_{HFE} = f(HEP_{nominal} \mid PSFs) \tag{4}$$

where:

- $HEP_{HFE}$ is the human error probability for the human failure event,
- $HEP_{nominal}$ is the nominal or default HEP provided in the method, and
- $PSF$ is the set of performance shaping factors that is considered in the method.

Of course, different HRA methods have vastly different approaches to estimating HEPs, and not all methods will formally enlist nominal HEPs or PSFs. Conceptually, however, the point remains that the HEP is a function of a particular probabilistic approach given some context that affects operator performance. Given this simplified approach, once the HEP is calculated as a function of how PSFs modify the nominal HEP, it remains unchanged over the (time) duration of the task (see Figure 4).

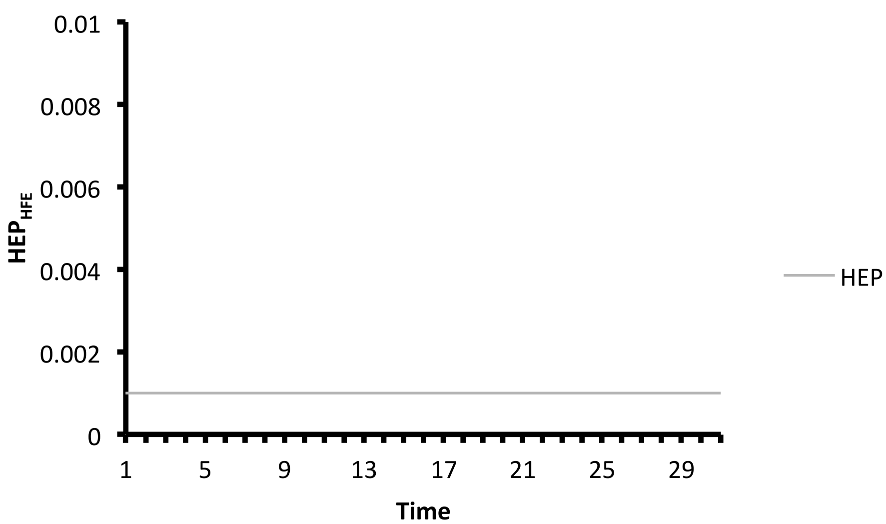


Figure 4: The non-effect of time on the error estimate in static HRA

It should be noted that SPAR-H does, indeed, model time as a PSF. Specifically, SPAR-H analyzes the impact of available time to complete the task on the HEP. A shorter window of time degrades the operator's performance or at least their ability to complete the task successfully. The modeling of time as a PSF is, however, not the same as dynamic HRA. Time, as modeled in SPAR-H and other HRA methods, is dynamically invariant for the HFE. For the specific HFE being analyzed, the analyst will not typically look at a range of time windows or how that time window changes throughout alternate event evolutions. Time, in static HRA, is simply a snapshot of an available resource the operator needs, which is firmly fixed in a predefined HFE in the PRA.

## 2.3 Extending the Current Paradigm: Dynamic HRA

The preceding discussion has centered on HFE modeling and HEP quantification for conventional HRA, which is static in nature. Once the overall system is modeled, including HFEs, they do not change as a result of the event progression. Dynamic HRA does not rely on a fixed set of event and fault trees to model event outcome. Rather, it builds the event progression dynamically, as a result of ongoing actions (Acosta & Siu, 1993). The dynamic approach in PRA has proved especially useful for modeling beyond design basis accidents, where not all failure combinations (and, importantly, not all recovery opportunities) can be anticipated or have been included in the static model. Additionally, the failure of multiple components or unusual sequences of faults, even within design basis, may challenge the fidelity of the PRA model. While such events are rare, dynamic modeling affords the opportunity to anticipate such permutations and address them in a risk-informed manner should they occur.

Boring (2007), among others, explains the conceptual shift from static HRA to dynamic HRA. Key aspects of this shift are the transition from predictions based on fixed models of accident sequences into predictions based on direct simulation of an accident sequence, with explicit consideration of timing of key events. For HRA to fit into this dynamic framework, the models must follow a parallel path, shifting away from estimating the probability of a static event, and into simulating the multitude of possible human actions relevant to an event.

Traditional static HRA attempts to directly estimate or assign probabilities to defined HFEs. Example HFEs are “failure to initiate feed and bleed” and “failure to align electrical bus to alternative feed.” In this new dynamic HRA framework, the focus shifts to simulating the human performance within a dynamic PRA framework and using the results of those simulations to assign the HEP. Instead, dynamic HRA yields HFEs such as “failure to initiate feed and bleed over time.”

In essence, the HEP that is quantified varies over time as PSFs change in their influence:

$$HEP_{dynamic} = f(HEP_{nominal} \mid PSF(t)) \tag{5}$$

where $t$ is time. The PSFs change their influence on the HEP over time, because the PSFs change states.

This dynamic formulation of the HEP in Equation 5 is similar to the static formulation in Equation 4 in that the HEP is quantified as a function of the nominal HEP as adjusted by PSFs. The key difference is that both the state of the PSFs and the influence of the PSFs can change over time. The final effect is that the HEP varies over time (see Figure 5).

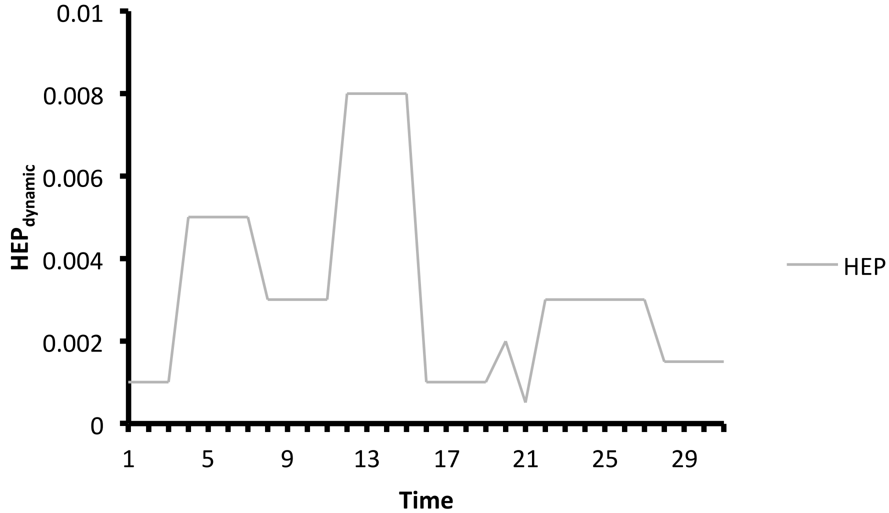


Figure 5: The effect of time on the error estimate in dynamic HRA

Dynamic HRA promises opportunities to model event progressions and outcomes beyond what's possible with static PRA models. As depicted in Figure 6, dynamic HRA can also provide an ongoing quantification of the HEP at any given point in time. Each subtask performed has an accompanying error rate, which can be combined with other subtask HEPs to form a joint HEP representing the entire HFE. The relationship between subtasks and time remains nonlinear. Subtasks require time, but each subtask will do so differently. As such, it is often convenient to consider the subtasks in terms of windows of time. Hypothetical Tasks A – I are parsed across the timeline in Figure 6. Within each subtask time window, there is an HEP. This subtask HEP may be represented as an averaged single-point subtask HEP across each time window or as a function representing the distribution of the HEP within each subtask (see Figure 7). Additional information such as the uncertainty quantification may also accompany each subtask HEP.

Note that the overall HEP cannot be calculated before the entire HFE has been modeled. Even though dynamic HRA does not require a predefined event tree, it must model all relevant subtask outcomes to arrive at the overall HFE. Dynamic generation of subtask HEPs does not result in joint HEPs until all subtasks in the HFE are modeled.

In adapting HRA from static to dynamic modeling, there are three essential considerations. First, the dynamic HRA approach advocated by Boring (2007) relies on PSFs to capture operator performance. Negative PSFs serve to increase the HEP over a nominal rate, whereas positive PSFs decrease the HEP over a nominal state. For example, the stress PSF may serve to increase the HEP, while crediting the procedures PSF may decrease the PSF. As discussed in Boring (2007), some PSFs remain constant across an event progression, while others change (see Table 1). Some PSFs may change gradually, while others may change suddenly as a result of rapid changes in the plant or individual. Errors are driven by PSFs. In this context, the error propagation is not a result of the presence of an HFE yielding overall increases in subsequent HFEs. The gradation of human performance is modeled through PSFs, and those PSFs have

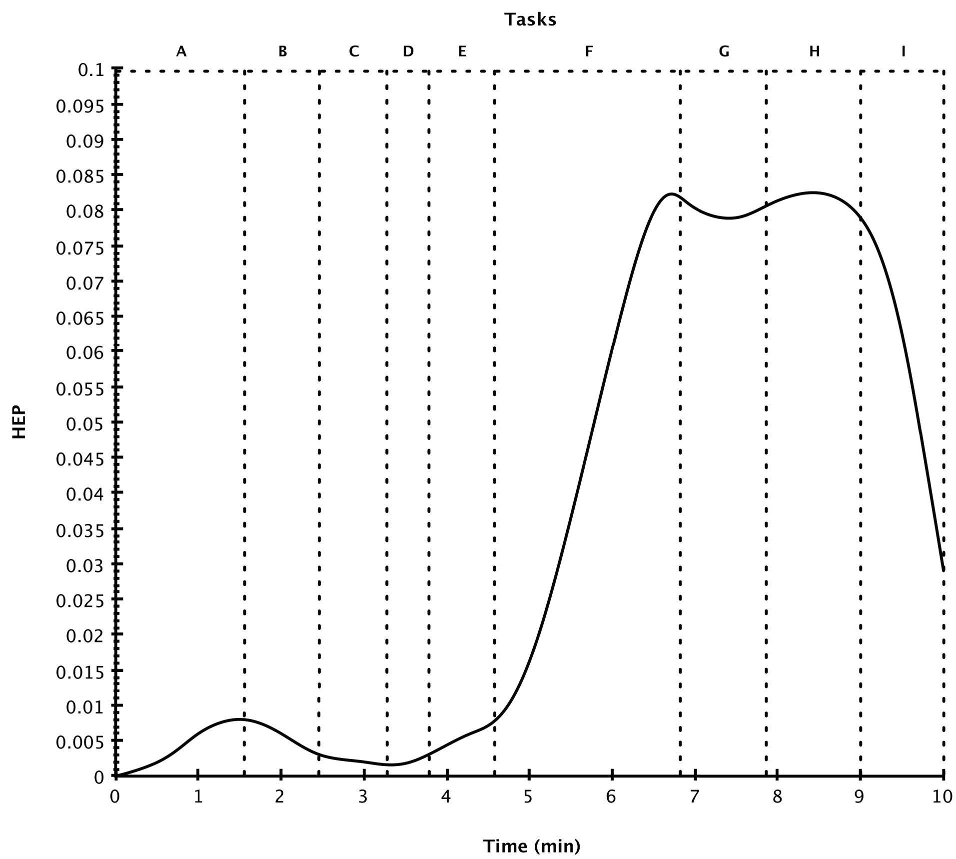


Figure 6: Hypothetical subtask HEP calculation for a dynamic event progression

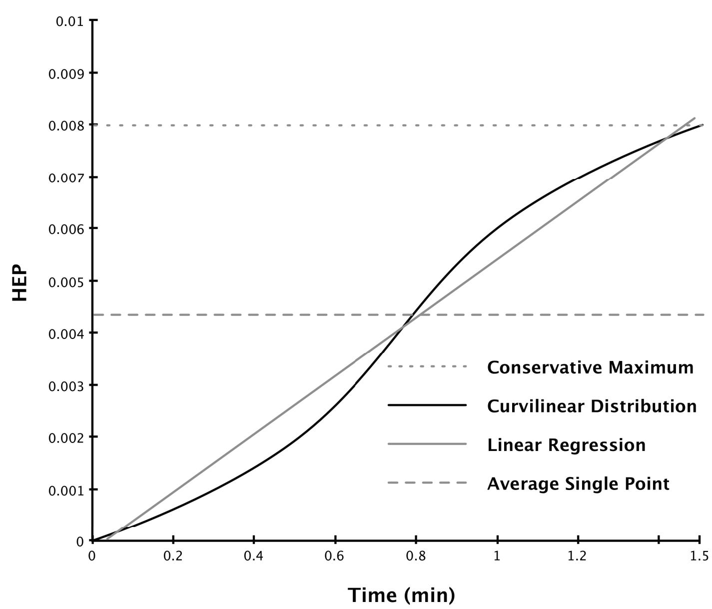


Figure 7: Four types of subtask HEP estimation

Table 1: Types of PSF modifications

| Static Condition | Dynamic Progression | Dynamic Initiator |
|---|---|---|
| *PSFs remain constant across the events in a scenario.* | *PSFs evolve across events in a scenario.* | *A sudden change in the scenario causes changes in the PSFs.* |

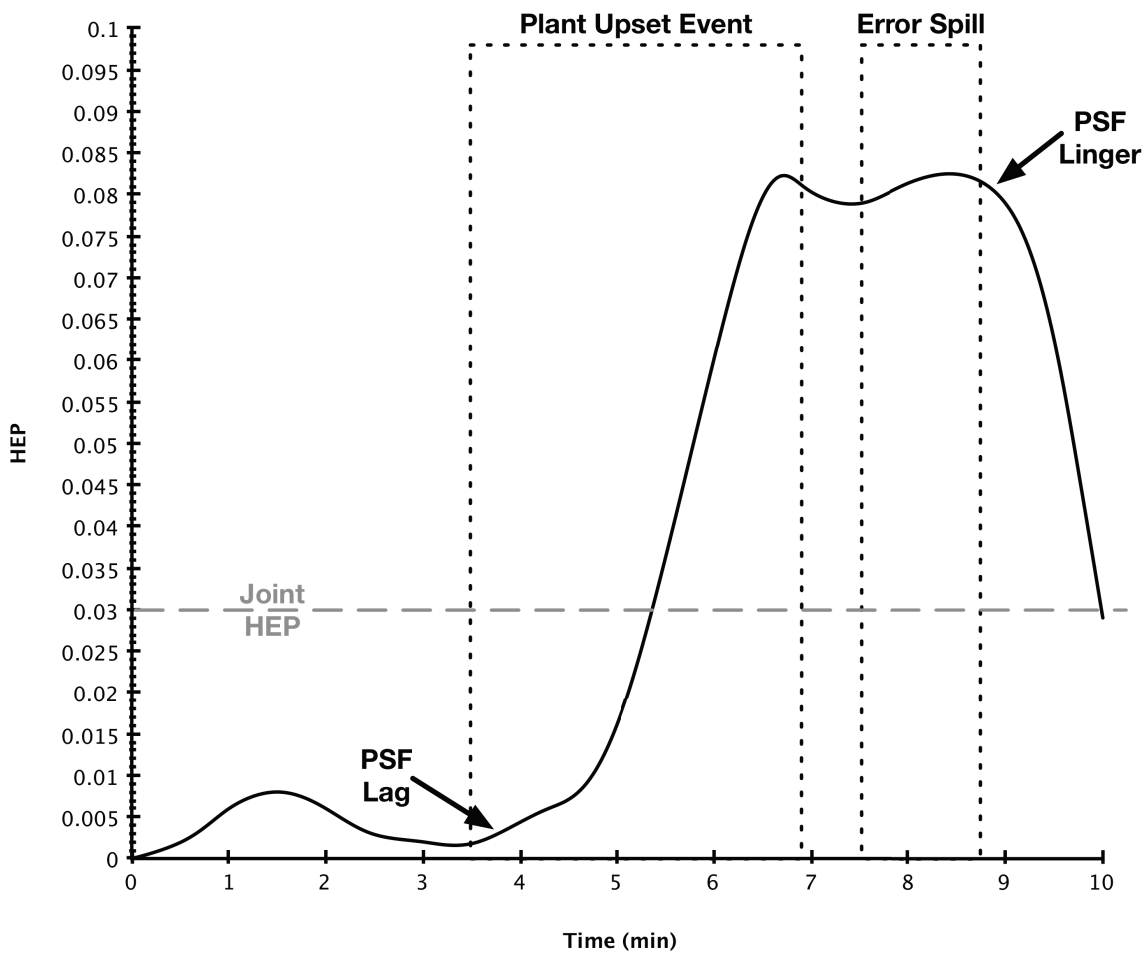


Figure 8: Illustration of dynamic HRA considerations

influence across subtasks and even, in some cases, across HFEs. Even though one event may yield overall successful performance, the degraded state of particular PSFs may drive the error likelihood of events later in the sequence.

Second, PSFs have qualities of *lag* and *linger*. Rarely is a PSF (such as stress) instantly invoked. Rather, it builds up over time, even after the initiation of a plant upset event. Some PSFs may have a gradual onset, while others may have more immediate effects. As shown in Figure 8, there is a delay in the increase of the subtask HEP after the initiating event. In this case, it represents *PSF lag*, by which the operator does not immediately psychologically or physiologically respond to the event. Likewise, once a particular PSF is manifest, it may not diminish instantly. For example, stress may accumulate, and it may take considerable time for the effects of stress to dissipate, even after the trigger of the stress has subsided. This is illustrated in Figure 8 as *PSF linger*, whereby the elevated HEP continues into subsequent subtasks, even after the plant event has terminated. These two qualities—lag and linger—are not currently considered in HRA methods that use PSFs. Yet, to model the effects of PSFs, it is not simply a matter of identifying the discrete effects of a particular PSF on performance at one point in time. The effects of PSFs must be considered temporally, as the PSFs will have a range of effects across the event sequence. Subtasks should never be analyzed in isolation. They must always consider the antecedent PSF context, which may lag or linger to produce dependent effects.

Finally, there is the concept of *error spilling*. When an error occurs, it often has effects downstream. Similarly, when PSFs are activated, they not only have temporal effects but also lateral effects on other PSFs. It is well understood in HRA that many PSFs are not independent from each other (Groth & Mosleh, 2009). PSFs are, in fact, entangled, and the effects of one PSF will tend to spill over into other PSFs. For example, task complexity will invariably affect the workload and stress of the individual performing the task. This error spilling between PSFs has been largely unaccounted for in HRA modeling. It can best be understood as an emergent property that should be modeled dynamically. Error spilling is manifest in Figure 8 as a hypothetical surge in the subtask HEP after the initial plant upset event subsides. While such a surge could simply be the result of a PSF linger episode, it is likely that conflation across PSFs would serve to disrupt the operator's performance and recovery from the event.

## 2.4 A New Conceptualization of HRA: Computation-Based HRA

As previously mentioned, one goal under the RISMC pathway is to include robust and dynamic models of human performance, or computation-based HRA methods, in the RAVEN simulation framework. In the context of the RISMC pathway, computation-based HRA methods are not just updated dynamic HRA methods, but also have a spatial dimension, include mechanistic codes, and factor in the topology of the problem space. By problem space, we mean the domain defined by the relevant parameters that define the domain's boundaries and define the relevant issues or factors within that space. For example, the problem space for static HRA includes parameters such as the nominal HEP and PSFs. The spatial dimension refers to the notion that some risk significant events at plants, in particular external events, have important spatial or geographic aspects to their problem space. Fires, floods, earthquakes, and other natural disasters affect multiple structures and systems on the physical grounds of the nuclear power generating station (e.g., the main control room, the switch yard, emergency diesel generators, turbine building, etc.). Most PRAs and HRAs currently do not model these spatial aspects of external events. The fact that the RISMC toolkit explicitly merges mechanistic thermohydraulic multi-physics codes with probabilistic risk models, and dynamic HRA methods do not, also means that it is more appropriate to call what is being developed in this research effort as *computation-based HRA*.

The RISMC framework also considers the topology of the problem space, and explicitly models its characteristics (e.g., its features, boundaries, and the nature of the interactions between relevant parameters). The complexity of the interactions between relevant parameters can be simplistically represented by the number of interactions between "nodes" or parameters in the model. For example, if a problem space has four nodes, the complexity of the topology can range from linear, where the maximum complexity that can occur is one node interacting with its two adjacent nodes, to fully crossed, where every node interacts with every other node. The difference in these two topologies is depicted in Figure 9.

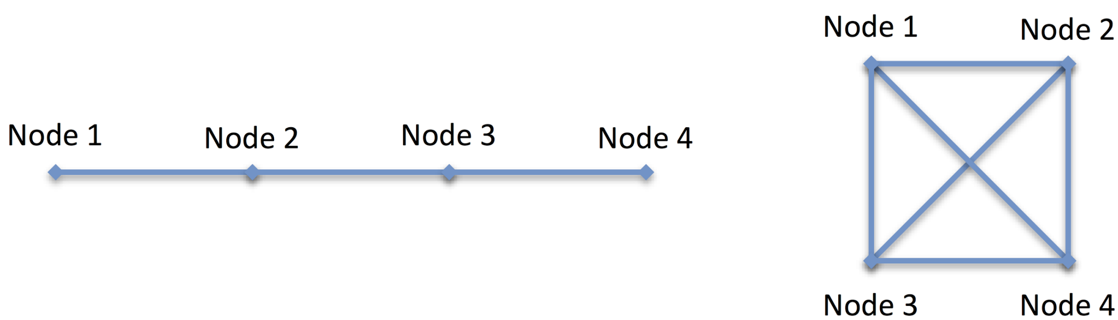


Figure 9: Range of variability in topology/complexity of four hypothetical nodes

More concretely, the differences between the topology of static HRA's problem space and computation-based HRA's problem space can be seen in the following formulas:

$$HEP_{HFE} = f(HEP_{nominal} \mid PSFs) \quad (6)$$

$$HEP_{Computational} = f(HEP_{nominal} \mid PSF(t,s)) \quad (7)$$

where $s$ is space.

In static HRA, parameters such as time and space are not considered in the problem space, whereas computation-based HRA's topology does take them into consideration. What needs to be further elucidated in computation-based HRA is the nature of the relationships between the nominal HEP, PSFs, time, and space parameters.

Achieving the RISMC goal of including computation-based HRA methods in the toolkit will require additional research and development (R&D), data collection, and validation activities. Based on the above dimensions or factors that constitute computation-based HRA, it is clear that there are a variety of reasons for this. First, as Boring, Joe, and Mandelli (2015) and Joe et al.

(2015) pointed out, widely used HRA methods have been and remain static. They do not provide a dynamic account of human actions. Thus, it would not be feasible to simply take these methods in their current forms and insert them without any modifications into the RISMC toolkit.

Second, current HRA methods typically do not model external events (e.g., flooding). In traditional PRA nomenclature, the modeling of external events corresponds to Level 3 PRA. *Level 1* PRA concerns potential core damage, *Level 2* PRA concerns potential release of radioactivity (i.e., a severe accident), and *Level 3* PRA concerns potential consequences of a severe accident in terms of health and environment. For a variety of reasons, including the safety track record of the nuclear industry and the cost to develop a full plant PRA, events are not as well defined and modeled in Level 2 and Level 3 PRA as they are in Level 1. Consequently, when considering the quantification of human performance for Level 2 and Level 3 HRA, the technical basis for determining the nominal HEP is not well established. There is little operating experience available, and most of the widely used HRA methods were developed for Level 1 events. Some recent efforts are underway to include HRA in Level 2 PRAs (Boring et al., 2015), but the fact that RISMC has the goal to model and quantify external events means that the HRA methods developed under Level 1 PRA assumptions should not necessarily be used in their existing form in the RISMC toolkit. At a minimum, the existing HRAs' underlying assumptions and their methodologies need to be revised to consider the expanded scope and complexity of external events, and if the R&D efforts to update existing methods proves infeasible, new HRA methods specifically designed to address Level 3 PRA issues will need to be developed.

Third, with the exception of the efforts to merge thermohydraulics with HRA (see Chapter 3 for discussion), the field of HRA has yet to take existing HRA methods and marry them dynamically to deterministic multi-physics codes. Lessons learned and best practices from this past work will be leveraged by this RISMC effort, but this R&D will also benefit from advantages that were not available to past efforts, including better integrated programming languages, updated thermohydraulic codes, and state-of-the-art supercomputing.

And finally, while some aspects of the topology of the problem space has been considered in HRA, such as dependence, many of the simplified HRA methods employed today use dependence calculations sparingly, as it appears a considerable amount of the dependency in tasks/events is accounted for by expert judgment when identifying and framing the risk significant human errors and successful human actions.

The conceptual shift from static to dynamic to computation-based HRA can be best explained by analogy. If, for instance, we are interested in building a model for coin flips:

- One approach would be to focus on estimating how often we see heads vs. tails. Within this approach, the coin-flipping analyst would directly assign a probability via experimentation. For example, with no preconceived notion of the outcome, the analyst would best resort to empirical observation. He or she might try flipping the coin 100 times, counting the heads and tails, and determining the ratio. A fair coin would result in approximately 50 heads and 50 tails. Given sufficient statistical power and confidence in the validity of the results, the analyst might form a probabilistic model to account for the model of the coin flip. This data-based approach to modeling the coin flip would be brittle, however. For an unfair weighted coin, for example, the analyst might see something else like 25 heads per 100 flips. In another case, a human could intervene with the coin flipping process (e.g., by trying to grab the coin out of the air) resulting in 30 heads per 100 flips. The simple model would have difficulty accounting for these complexities. .

- A second approach would be to simulate what happens as the coin flips. This simulation can be repeated many times, and the results can be aggregated into a probability. The clever analyst accounts for additional factors at play in the coin flipping and makes adjustments to the predictions. For example, the model could make an adjustment for the effect of the weight of the coin, and thus predict that the biased coin would fall more consistently heads down (60% of the time) whereas a fairly weighted coin would fall evenly (50% heads, 50% tails). This model still wouldn't be able to account for the human attempting to grab the coin.
- A third approach is to model how the coin flips. Here, the analyst simulates the physical processes involved in the coin flip, to repeat that simulation a large number of times, and to then incorporate the results of the simulation set into a probability. The explanation of the mathematics of this approach is beyond the scope of this paper, but the benefit is clear: the simulation of the physical process can be used on any coin, under a range of conditions. The simulation model can be combined with other models (e.g., of the weather conditions while the coin is being flipped). Only through simulation do we have the power to predict the results of the coin flip under a wide range of conditions, including human interventions such as grabbing at the coin or changing the weight of the coin. There is considerable extra complexity involved in this third case—it represents a thorough understanding of the properties underlying the coin throw, the gravitational pull on the coin, the properties and behaviors of the coin itself, and perhaps factors like the ambient temperature of the air, air flow, or even the terrain where the coin will land. Yet, this modeling approach offers a more complete representation of coin flipping—a representation that is adaptable and flexible to changes in the modeling assumptions.

So it is with HRA. There are multiple ways of approaching the problem of predicting operator performance. The static approach provides a good general purpose statistical representation of the outcomes of operator actions in generic situations under stable, known conditions. The static model assumes that the operators have nominal error rates, and there is some mathematical treatment to account for factors that can influence operator performance. In turn, the dynamic approach accounts for more factors, notably time. Analysts understand how performance can change over time and model the permutations of those changes in terms of consequences to the overall HRA. The dynamic approach, while better accounting for the spectrum of performances that are possible, does not actually attempt to provide a high fidelity model of operator performance. It may model the range of behaviors, but the outcomes are known, and the effects are mostly treated as mathematical adjustments to performance curves over time. In the computational approach, there is an attempt to interface a realistic model of the operator—a virtual operator—with a realistic plant model. The attributes that shape performance are known, and the functions of human performance are accounted for. What computation-based HRA affords the analyst is removal from pre-scripted scenarios. It is possible for the computational model to account for unexampled events and to respond in a realistic manner to the evolution of the plant. Static HRA provides a subset of outcomes; dynamic HRA adds changes over time to understanding a wider range of outcomes; computation-based HRA affords the analyst the chance to see the emergence of behaviors interacting with the plant, encompassing a more complete cross-section of reality.

This page intentionally left blank

# 3. A COMPUTATIONAL ENGINE FOR HUMAN RELIABILITY ANALYSIS

## 3.1 Introduction

Computation-based HRA requires both a dynamic HRA method and a computational "engine" to drive its modeled interaction with the plant. Boring et al. (2015) and Joe et al. (2015) previously summarized current approaches to human performance modeling (HPM). HPM represents cognitive models that simulate aspects of human perception and decision making. These approaches are akin to artificial intelligence, although they tend to be focused on modeling the specific processes of cognition rather than the general knowledge associated with humans. In other words, they are problem solving algorithms that reasonably approximate human cognition. To date, few HPM approaches have been applied to HRA. Mainstream HPM approaches mimic cognition but not necessarily in terms of probabilistic performance. An HPM approach might be a good predictor of behavior, but it may not be good at predicting multiple outcomes, ranking their likelihood, nor detailing failed performance.

Thus, it is essential that computation-based HRA have a computational engine to drive the interaction of the HPM with the system. The role of the computational engine is three-fold:

- It interfaces the HPM's virtual operator with the virtual environment. In a nuclear power context, the computational engine represents the plant model (e.g., a RELAP (David et al., 2012) thermohydraulics model of the systems comprising a specific nuclear power plant). The plant is not limited only to thermohydraulics models but also includes other systems like electricity generation and the human-machine interface in the control room. The computational engine integrates these disparate plant elements into a single model with which the virtual operator can interface.
- The computational engine serves as the scheduler for event progression. The sequencing of plant events and the interface between the virtual operator and those events should be automated. For example, it may be desirable to investigate permutations of operator response (e.g., immediate vs. delayed response to an annunciator). Virtual operator responses can be automated in a scheduler to see the range of outcomes and emergent effects on the plant. Manually configuring and controlling such operator actions would prove needlessly labor intensive and time consuming. Depending on the capabilities of the underlying plant model, in some cases this scheduling may represent real-time processing, but it is often possible through optimized and parallelized codes to achieve faster-than-real-time processing. Faster-than-real-time processing makes possible on-the-fly HRA, including online risk monitoring and situation look-ahead.
- Since the computational engine is used in support of PRA, it may automate many of the probabilistic calculations. For example, it may be used to compute the core damage frequency over an entire event tree automatically. It may also be used to calculate the HEP of a particular set of human actions. When used for Monte Carlo style simulations, it may also be used to determine the probability distribution or limit surface for particular operator-influenced outcomes. In order to accomplish these tasks, the computational engine should serve as the data historian, logging key plant parameters, transients, and

operator actions. This data logging capability is a minimum requirement; however, additional options like providing probability functions can serve to increase the usefulness of the engine in support of PRA and HRA.

These three roles—integration of plant and operator models, event sequencer, and probabilistic toolset—are desirable, but they are not all prerequisites to the computational engine. In this chapter, we discuss several candidate computational engines to determine their feature sets and suitability as a component of the RISMC computation-based HRA framework.

## 3.2 Candidate Computational Engines

### 3.2.1 Review of the ADS Approach

The Accident Dynamic Simulator (ADS) framework (Chang & Mosleh, 2007a and 2007c) was developed by the Center for Risk and Reliability at the University of Maryland. The framework addresses the limitations of classical PRA methods, which are based on fault tree and event tree logic structures, in handling dynamic and time-dependent interactions between system elements, physical processes, and human operators.

Compared to similar dynamic PRA codes, the ADS framework features the integrated Information Decision and Action in Crew Context (IDAC) cognitive model, designed to perform automatic, systematic, and probabilistic simulation of human–system interactions. IDAC consists of six modules (see Figure 10) that model specific components of such human–system interactions:

1. *Crew module:* modeling crew response
2. *System module:* modeling system response
3. *Indicator module:* modeling the control panel inside the control room
4. *Hardware reliability module:* modeling possible system failures and effects
5. *Scheduler module:* controlling simulation sequence
6. *User interface module:* facilitating interaction between the analyst and the software

The ADS framework uses a Dynamic Event Tree (DET) methodology (Amendola & Reina, 1984) to model accident sequences following an initiating event. Branches are generated at discrete points in time, based on probable alternative outcomes resulting from changes of system and operator states. Currently, Dynamic Decision Event Tree (DDET) branches are generated due to the operators' cognitive activities and actions, and hardware failure.

Accident evolution is calculated at each time step $\Delta t$; at the beginning of each $\Delta t$ the system configuration is updated and if a branching condition is met, a new set of branches are generated as shown in Figure 10 (Chang & Mosleh, 2007a). Processes within the ADS framework include the following:

1. System Module first calculates the system state to the next time step (from $t$ to $t + \Delta t$).
2. Indicator Module updates its indications to reflect the new system state.

3. Hardware Reliability Module detects the state change in System Module and Indicator Module, calculates the probabilities of hardware failure. New sets of branches may be generated in this step.
4. Crew Module models the operators' reactions who respond cognitively, emotionally, and/or physically to the new situation. New sets of branches may be generated in this step.
5. Scheduler Module generates any needed event tree branches and calculates branch probabilities based on branches information generated in (3) and (4).
6. Scheduler Module determines whether the current sequence should be terminated.
7. If current sequence is terminated, then go to (1) to start a new cycle of simulation to next time step. Otherwise Scheduler Module searches for a branch that has not reached scenario termination criteria and restarts the simulation from that point. The process repeats itself until all branches are simulated.

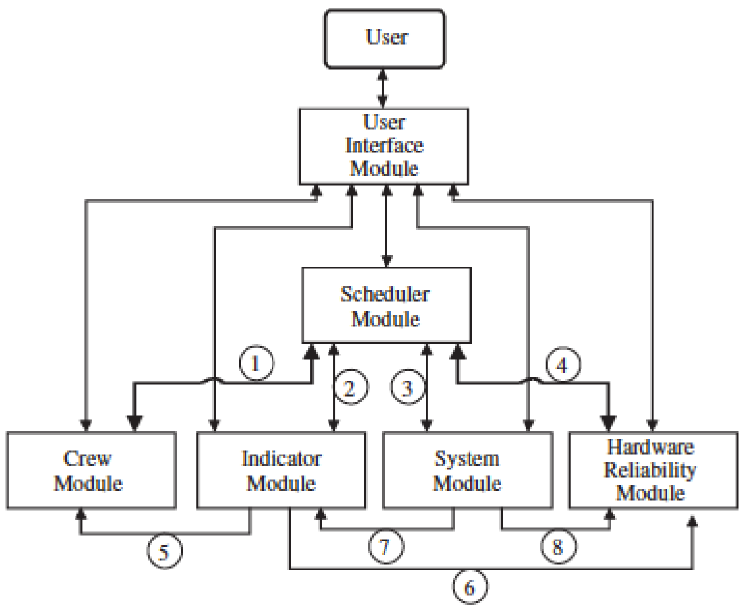


Figure 10: Overview of the ADS framework [D9]

For the scope of this report we are interested in summarizing the capabilities to model human behavior during the accident scenario. As mentioned earlier, the IDAC module (Chang & Mosleh, 2007c) is actually in charge of performing such modeling. An overview of the IDAC operator response module is shown in Figure 11 (Chang & Mosleh, 2007c):

- IDAC implicitly implements the decision process of the operator (outgoing action) as a function of the status of the system (incoming information)
- Incoming information passes through an external filter which can block/distort the information coming from the outside world
- The information passing through the external filter is then processed by the operator model, which is modeled by three sub-modules:

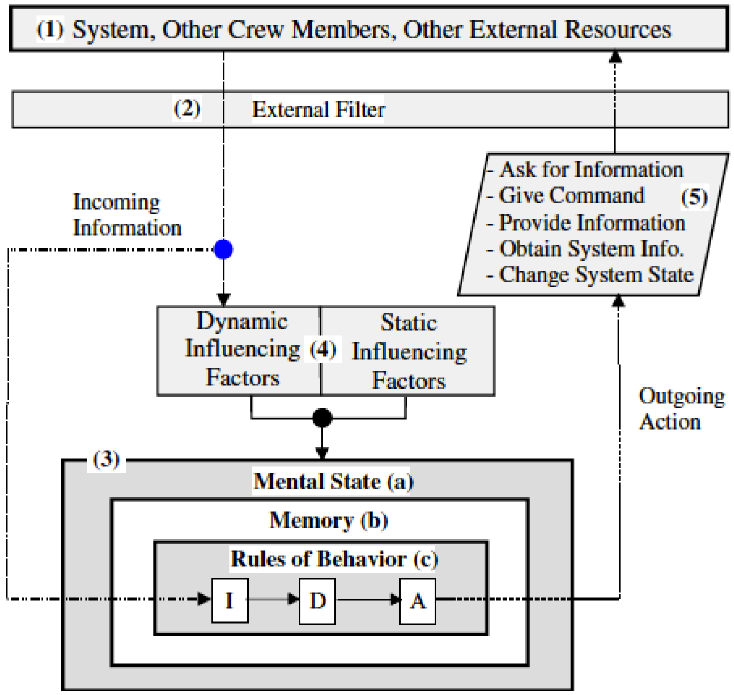


Figure 11: IDAC dynamic response model (Chang & Mosleh, 2007c)

- *Mental State:* defined by a set of performance influencing factors (PIFs). This defines the operator's state of mind in various dimensions such as individual differences, situation perception and appraisal, feelings about the situation, and certain cognitive behavioral modes (e.g., bias).
- *Memory:* three types of memory are modeled here:
  - Working Memory stores limited information related to the current cognitive process.
  - Intermediate Memory, theoretically unlimited in capacity, stores information related to recent cognitive processes that could be easily retrieved at any time given appropriate stimuli.
  - Knowledge Base, also theoretically unlimited in capacity, stores all problem-solving and decision-making-related knowledge obtained from training and experience.
- *Rules of Behavior:* Rules of Behavior governs the cognitive, emotional, and physical responses of an individual for a given state of PIFs and the content of memory. More specifically, the migration of memory and mental state from one state to another during the course of an event, as well as corresponding operator behavior in the I–D–A sequence are regulated by the Rules of Behavior.

- Actions are external manifestations of decisions (to act) formed by the cognitive processes of problem-solving and decision-making. Through action the operators interact with the external world, which in turn generates new information starting another I–D–A cycle. The operator's actions could be blocked or distorted by the external filter. This interaction loop continues until a certain system state is reached (e.g., problem solved or an undesired system state reached).

### 3.2.2 Review of the ADAPT Approach

The Adaptive Dynamic Accident Progression Trees (ADAPT) (Hakobyan et al., 2008) is a software tool developed by The Ohio State University in collaboration with Sandia National Laboratories. This tool is designed to perform forward uncertainty propagation of both aleatory and epistemic uncertainties on a dynamic stochastic system. Its main range of applicability focuses on the analysis of nuclear power plant accident scenarios (Level 1, Level 2 and Level 3 PRA (NRC, 1990)) but can be extended to any type time-dependent analysis.

The statistical calculation engine is based solely on the DET approach. The DET approach is a methodology that couples an event-tree based logic approach with system simulator codes, e.g., RELAP5-3D and Methods for Estimation of Leakages and Consequences of Releases (MELCOR). The user input information, along with the desired system simulator code, contains a set of branching and stopping rules. Each rule specifies when the simulation splits into two branches, with each branch in the system configuration modified accordingly to the branching rule.

This coupling allows the user to simulate several accident scenarios in considerably less time than a classical Monte Carlo approach since the fraction of the simulation run that has been modeled in previous runs is not repeated. This advantage is implicitly gained in the construction of a tree-based data structure where a fraction of the simulation run is performed in each branch.

The branching conditions require two types of information:

- Event type, and
- Cumulative distribution function (CDF) associated with the event.

An example is given in Figure 12 (Hakobyan et al., 2008) which shows the CDF associated with the rupture event of a PWR system surge line as a function of the creep rupture parameter $R$. In this case the CDF $\Phi(R)$ is sampled five times (0.05, 0.25, 0.5, 0.75, 0.95) with each sample representing a branching condition. The analysis follows this path:

1. A single simulation run is launched with the value of $R$ generated by the simulation constantly monitored; the branching condition is set to the first CDF sample point $\Phi(R) = 0.05$
2. When a value of $R = 0.518$ is reached ($\Phi(0.518) = 0.05$), the first branching condition is met. At this point the simulation is stopped and two simulation branches are generated:
    a. One simulation continues where a break in the surge line is added
    b. A second simulation continues the original simulation run with the branching condition updated to the new value $\Phi(R) = 0.25$
3. Step 2 continues until the last branching conditions is reached

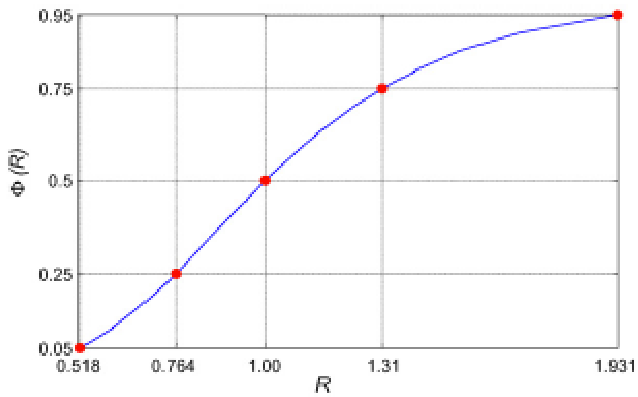


Figure 12: Example of CDF discretization in an ADAPT DET approach

Human related stochastic events are modeled in exactly the same way by providing a CDF for each human-related event. An example is given in Winningham et al. (2009) where ADAPT is linked to a RELAP5 model (2012) of the Reactor Vessel Auxiliary Cooling System (RVACS) passive heat removal system in a sodium cooled fast pool-type reactor with metallic fuel. In Winningham et al. (2009), the scope of the analysis is to assess the probability of recovery of the RVACS system following an aircraft crash in time to prevent fuel damage. The RVACS recovery is performed by a recovery crew and, as part of the analysis, uncertainties are associated to the crew arrival time and recovery time of each of the RVACS cooling towers (see Figure 13).

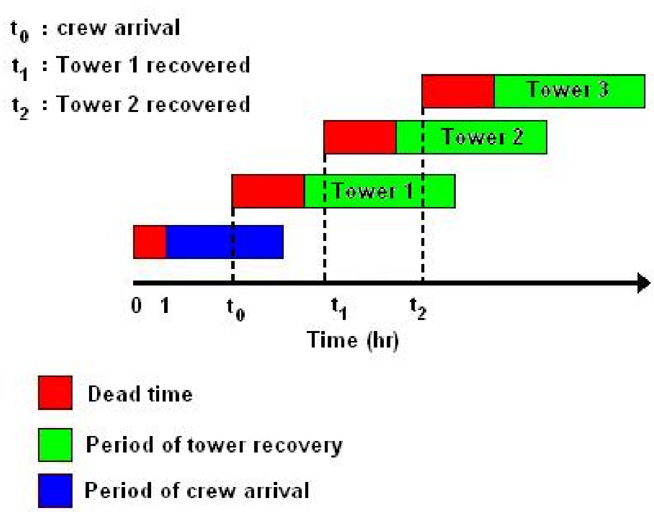


Figure 13: Events associated with the crew recovery arrival time (blue bar) and RVACS tower recovery times (green bars) (Winningham et al., 2009)

### 3.2.3 Review of the MCDET Approach

The Monte Carlo and Dynamic-Event Trees (MCDET) method (Kloos & Peschke, 2006) has been developed by Gesellschaft für Anlagen- und Reaktorsicherheit (GRS), the German federal nuclear regulator. It blends both the Monte Carlo (MC) and DET methods to analyze accident scenarios. The key issue of this approach is the concept of “transition” which is defined by the tuple:
Transition = (when, where to)

Both elements of this tuple can be of different types, i.e., they can be:

- Deterministic
- Random
    - Continuous
    - Discrete

The MCDET is designed to handle all possible combinations of element types as shown in Table 2.

Table 2: MCDET elements modeling combinations

| | | *Transition element* | |
|---|---|---|---|
| | | When | Where to |
| *Type* | Deterministic | Code | Code |
| | Random – Discrete | DET | DET |
| | Random – Continuous | Monte-Carlo | Monte-Carlo |

The MCDET methodology consists of several different routines that perform a specific task; the most relevant are the following:

1. *Initialization of the epistemic variables:* sets the values of the epistemic variables
2. *Initialization of the system conditions and of the aleatory variables:* performs initialization of the aleatory variables
3. *MC simulation:* detects conditions that require MC simulation and generates the random numbers for the aleatory variables concerned. If routine 3 detects an absorbing state, i.e. damage states, the states of no damage and controlled operation, or the specified end of the processing time it provides the corresponding information, and the calculation of the current sequence may be terminated.
4. *DET simulation:* checks the system conditions of the current dynamics calculation. If routine 4 detects conditions that require the generation of a branching, it determines all branches that branch off the sequence currently calculated and stores them if the respective sequence probabilities at the branch point exceed a user-defined threshold.

### 3.2.4 Review of the RAVEN approach

RAVEN (Rabiti et al., 2014) is a software framework that acts as the control logic driver for the Thermohydraulic code RELAP-7, a newly developed software at INL. RAVEN is also a multi-purpose PRA code that allows the dispatching of different functionalities. The framework is designed to derive and actuate the control logic required to simulate both plant control system and operator actions (guided procedures) and to perform both Monte Carlo sampling (Zio et al., 1998) of random distributed events and DET-based analysis (Amendola & Reina, 1984). RAVEN consists of two main software components, which are discussed in the next two subsections:

- Simulation controller, and
- Statistical framework.

#### *3.2.4.1 RAVEN Simulation Controller*

One task of RAVEN is to act as controller of the RELAP-7 simulation while simulation is running. Such control action is performed using two sets of variables:

- *Monitored variables:* set of observable parameters that are calculated at each calculation step by RELAP-7 (e.g., average clad temperature), and
- *Controlled parameters:* set of controllable parameters that can be changed/updated at the beginning of each calculation step (e.g., status of a valve—open or closed—or pipe friction coefficient).

The manipulation of these two data sets is performed by two components of the RAVEN simulation controller:

- *RAVEN control logic:* the actual system control logic of the simulation where, based on the status of the system (i.e., monitored variables), it updates the status/value of the controlled parameters, and
- *RAVEN/RELAP-7 interface:* in charge of updating and retrieving RELAP-7 component variables according to the control logic.

A third set of variables, i.e. auxiliary variables, allows the user to define simulation-specific variables that may be needed to control the simulation. From a mathematical point of view, auxiliary variables are the ones that guarantee the system to be Markovian (Gardiner, 2002), i.e., the system status at time $t = \bar{t} + \Delta t$ can be numerically solved given only the system status at time $t = \bar{t}$.

The set of auxiliary variables also includes those that monitor the status of specific control logic set of components (e.g., diesel generators or alternating current buses) and simplify the construction of the overall control logic scheme of RAVEN.

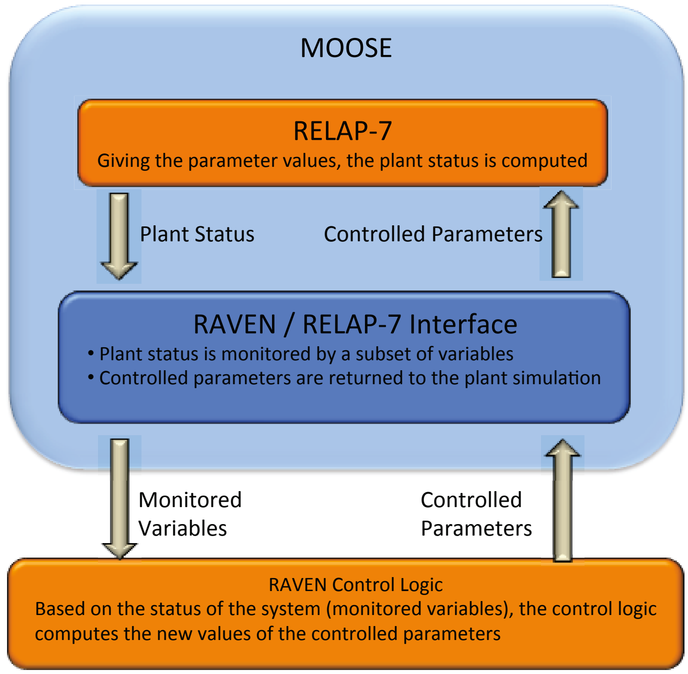


Figure 14: RAVEN simulation controller scheme

#### *3.2.4.2 RAVEN Statistical Framework*

The RAVEN statistical framework is a recent add-on of the RAVEN package that allows the user to perform generic statistical analysis. The statistical analysis includes:

- Sampling of codes, either a stochastic code such as the MC code (Zio et al., 1998) or Latin Hypercube Sampling (LHS) code (Helton and Davis, 2003), or a deterministic code (e.g., grid and DET (Amendola & Reina, 1984),
- Generation of Reduced Order Models (ROMs) as referenced in Queipo et al.( 2005) which are also known as surrogate models, and
- Post-processing of the sampled data and generation of statistical parameters (e.g., mean, variance, covariance matrix).

Figure 15 shows a general overview of the elements that comprise the RAVEN statistical framework:

- *Model:* represents the pipeline between input and output space. It comprises both codes (e.g., RELAP-7) and also ROMs,
- *Sampler:* the driver for any specific sampling strategy (e.g., MC, LHS, DET)
- Database: the data storing entity, and
- *Post-processing module:* the module that perform statistical analyses and visualizes results.

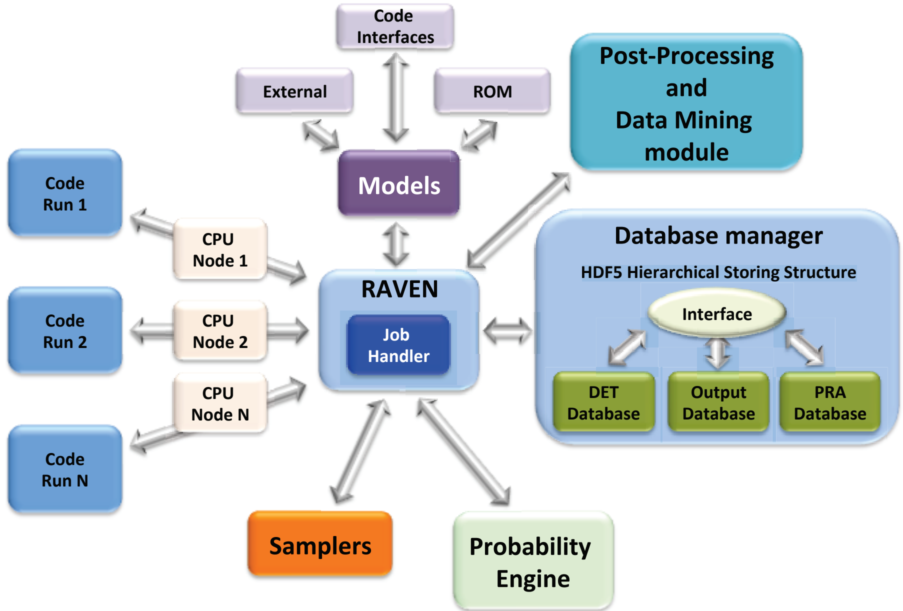


Figure 15: Scheme of RAVEN statistical framework components (Rabiti et al., 2014)

## 3.3 Suggested Path Forward

After reviewing the candidate computational engines, we have identified a set of requirements needed to perform computation-based HRA modeling:

- Tight coupling between system dynamics and human models;
- A model framework that couple external event, plant, and human models; and
- A more general statistical framework that uses multiple sampling strategies and data analysis tools.

From the list above we have determined that the RAVEN framework coupled with RELAP-7 best meets the requirements for the RISMC computation-based HRA framework. In particular, we have found it extremely valuable that it is possible to easily interface several HRA models to a single RAVEN-RELAP-7 simulation run. Such an interface is coded as a Python script, which allows integration of fairly complex models that can interact at each time step with the RELAP-7 simulation. In addition, the RAVEN statistical framework allows the user to perform a larger variety of stochastic analysis, ranging from classical MC and LHS strategies to more complex functions such as adaptive sampling and the creation of ROMs that can be, for example, embedded in the control logic itself.

# 4. A RESEARCH FRAMEWORK FOR DYNAMIC ELEMENTS OF HUMAN RELIABILITY

## 4.1 Introduction

In the previous chapter, we discussed the computational engine that binds thermohydraulic and other models with a dynamic scheduler and probabilistic inference toolset. Within the realm of hardware systems, this framework can accurately model plant performance. However, a significant influence on plant behavior and performance comes from the human operators who use that plant. The computational engine therefore needs to interface with a fourth element, namely a virtual operator that models operator performance at the plant. In current nuclear power plants (NPPs), most plant actions are manually controlled from the control room by reactor operators (ROs) or locally at plant systems by field operators. Consequently, in order to have a non-idealized model of plant performance, it is necessary to account for those human actions that control the plant. A high fidelity representation of an NPP absolutely requires an accurate model of its human operators.

While it is tempting simply to script human actions at the NPP according to operational procedures, there remains considerable variability in operator performance despite the most formalized and invariant procedures to guide activities (see NUREG-2127). Human decision making and behavior are influenced by a myriad of factors at and beyond the plant. Internal to the plant, the operators may be working to prioritize responses to concurrent demands, to maximize safety, and/or to minimize operational disruptions. While it is a safe assumption that the operators will act first to maintain safety and then electricity generation, the way he or she accomplishes those goals may not always flow strictly from procedural guidance. Operator expertise and experience may govern actions beyond rote recitation of procedures. As a result, human operators may not always make decisions and perform actions in a seemingly rational manner. Modeling human performance without considering the influences on the operators will only result in uncertain outcomes.

## 4.2 Dynamic HRA as a Possible Solution

Numerous dynamic HRA approaches have been developed (see Boring et al., 2014, for a summary). These approaches serve the role of the virtual operator. Dynamic HRA has, despite decades of research, been slow to take off. It could be argued that this is in part a byproduct of the limitations of computing power available to early modelers. For example, Rasmussen (1986) developed a cognitive model to be used in a complex computer program called the Dynamic Logical Analytical Methodology (DYLAM), which simulated PRA accident scenarios as a way to model and understand operator responses during emergencies in NPPs. Similarly, Wood, Roth, and Hanes (1986) and Woods, Pople, and Roth (1990) explored the feasibility of developing and using models of cognitive behavior in NPP personnel in simulations to improve the ability to predict human error during emergency operations. DYLAM, and the tools and techniques developed by Woods and his colleagues, namely Cognitive Environment Simulation (CES) and the Cognitive Reliability Assessment Technique (CREATE), likely suffered from the limitations of computation power in the 1980s.

In fact, computing limitations belie the fact that most dynamic HRA projects have been short-lived. Is this a consequence of the complexities of carrying out this research, regardless of the computing resources available? Perhaps the task of creating a dynamic HRA approach is no less difficult than the task of crafting an artificial intelligence system. Or, is the short-lived nature of these previous research products a consequence of the approaches not directly addressing research needs?

The goal within the RISMC framework is to ensure that previous hurdles to the success of dynamic HRA are not mirrored in computation-based HRA. As such, our approach is to ensure that research is undertaken in a systematic manner, with manageable proofs of concept serving as research milestones. Additionally, the research driving RISMC is not strictly for the purpose of advancing the science of HRA. Rather, the goal is to support other computational developments such as Multiphysics Object Oriented Simulation Environment (MOOSE; Gaston, Hansen and Newman, 2009) and RAVEN (Alfonsi et al., 2013). The human remains a seminal part of rich modeling of plant performance. RISMC strives to support this modeling through logical stepping stones toward comprehensive operator modeling.

## 4.3 A Computation-Based HRA Approach

The hybrid approach we propose for RISMC uses elements of human performance modeling and dynamic HRA found in existing methods. There are several criteria that we are adopting to guide the collection of different method elements:

- *Small number of PSFs:* Many existing dynamic HRA methods have been complex, e.g., in terms of the number of PSFs that are modeled. The RISMC approach should be simple enough to test proof of concepts. This means, a simplified PSF framework akin to the eight PSFs used in SPAR-H (Gertman et al., 2005) is a logical starting point.
- *Scalable:* The approach should lend itself to proof-of-concept demonstrations that can later be extrapolated to larger scale demonstrations. In this manner, the approach is iterative, allowing many tests to arrive at an optimal solution.
- *Not limited to time dynamics:* As noted, time is one of many variables that affect operator performance. These dimensions are not the same as PSFs—they are conditions that globally influence the PSFs. The approach should allow manipulation of these parameters in a systematic manner.
- *Simplified cognitive model:* As Coyne (2009) observes, it is not feasible to construct an entire artificial intelligence framework to drive the virtual operator. Such an approach would be fraught with the same difficulties and perils that have limited the field of artificial intelligence (see Section 5.2 for a discussion).
- *Sensitive to individual differences and crew performance:* Not all operators will respond identically in a given situation. Differences in performance are a simple byproduct of the characteristics of the individual operator (Joe and Boring, 2014). While most HRA methods attempt to arrive at a nominal or group normed performance, the RISMC approach will attempt to model these differences in order better to account for the range of possible outcomes. Likewise, there is a difference between individual operator and crew performance (Chang and Mosleh, 2007a), and the approach will seek to identify such nuances.

- *Able to make use of empirical data:* A final requirement is the ability of the approach to use available empirical data. Such data may drive performance distributions and also serve as the basis for quantification. The approach should allow incorporation of new data, e.g., the use of existing data as a prior with Bayesian updates (Groth, Smith, & Swiler, 2014). The approach should also identify where there are gaps in empirical data and guide future empirical research.

This broad approach to computation-based HRA is given the name *Human Unimodel for Nuclear Technology to Enhance Reliability* (HUNTER). HUNTER may be combined in a tongue-in-cheek manner with the computation engine RAVEN to create RAVEN-HUNTER or with the multiphysics toolkit MOOSE to create MOOSE-HUNTER. This term should not be taken literally, for the computation-based HRA framework certainly does not intend to stalk and eliminate any of the parent approaches. The hyphenated name should be considered the union of the two approaches.

The importance of the modeling approach, and particularly the cognitive models for decision making, is represented by the term *unimodel* as the “U” in HUNTER. The term unimodel appears primarily in the social psychological literature. Historically, research on persuasion suggested there were two types of processes—one effortful and thoughtful and one more intuitive or emotional. Such dual-route models are found in many psychological models, e.g., Kahneman’s slow vs. fast thinking (2011). Kruglanski and Thompson (1999) sought to unite the dual routes documented in persuasion research and proposed a unimodel to integrate them into a common framework. Here we borrow this concept of the unimodel—not as it applies to persuasion but, rather, as it can be used to indicate simplified models of cognition. The unimodel represents hybrid models of cognition that are required by HUNTER. These models are streamlined approaches to capture the general outcomes of operator decision making. HUNTER is a subset of possible human cognition in that it is coupled to the computation engine and provides a focused set of decision outcomes relevant to plant evolutions. These simplified models might be thought of as micro-models or even “smart” models. Just as a smart watch, smart television, smart phone, or other smart technological device does not seek to model the entirety of artificial intelligence, so it is in HUNTER. The HUNTER approach does not need to encompass all aspects and nuances of human cognition; rather, it seeks to implement the most relevant aspects for information gathering, decision making, or taking actions in a given operational context. The “smart” aspects of the operator incorporate relevant context such as PSFs needed to drive cognition but do not seek to model extraneous factors. The division between relevant and extraneous factors remains a topic for research, of course.

The following section outlines the main elements of the HUNTER approach that is being developed to interface with RAVEN.

## 4.4 Main Elements of the HUNTER Approach

### 4.4.1 Overview

The basic functionality of HUNTER is depicted in Figure 16 as an influence diagram. A plant starts at an initial state. The virtual operator (or crew) as modeled in HUNTER perform human actions to maintain or change the plant state, whereby modeling of the plant state is driven by the thermohydraulic and other models in RAVEN. The operator's actions are influenced by PSFs. HUNTER is event driven, and external events such as flooding have a direct effect on the PSFs that govern the operator's behavior. The operator actions, paired with plant changes, result in changes in the safety margin at the plant. Figure 17 presents a version of HUNTER over successive time steps, whereby this latter depiction is as a Dynamic Bayesian Network. The color-coding in Figure 17 shows that grey nodes represent elements coordinated through RAVEN (e.g., plant states, plant responses, and the safety margin) while the colored nodes are the human elements facilitated through HUNTER. These nodes may be compartmentalized such that the elements use aspects of separate HRA methods, e.g., the PSFs from SPAR-H (Gertman et al., 2005) and the human actions derived from IDAC (Chang and Mosleh, 2007b).

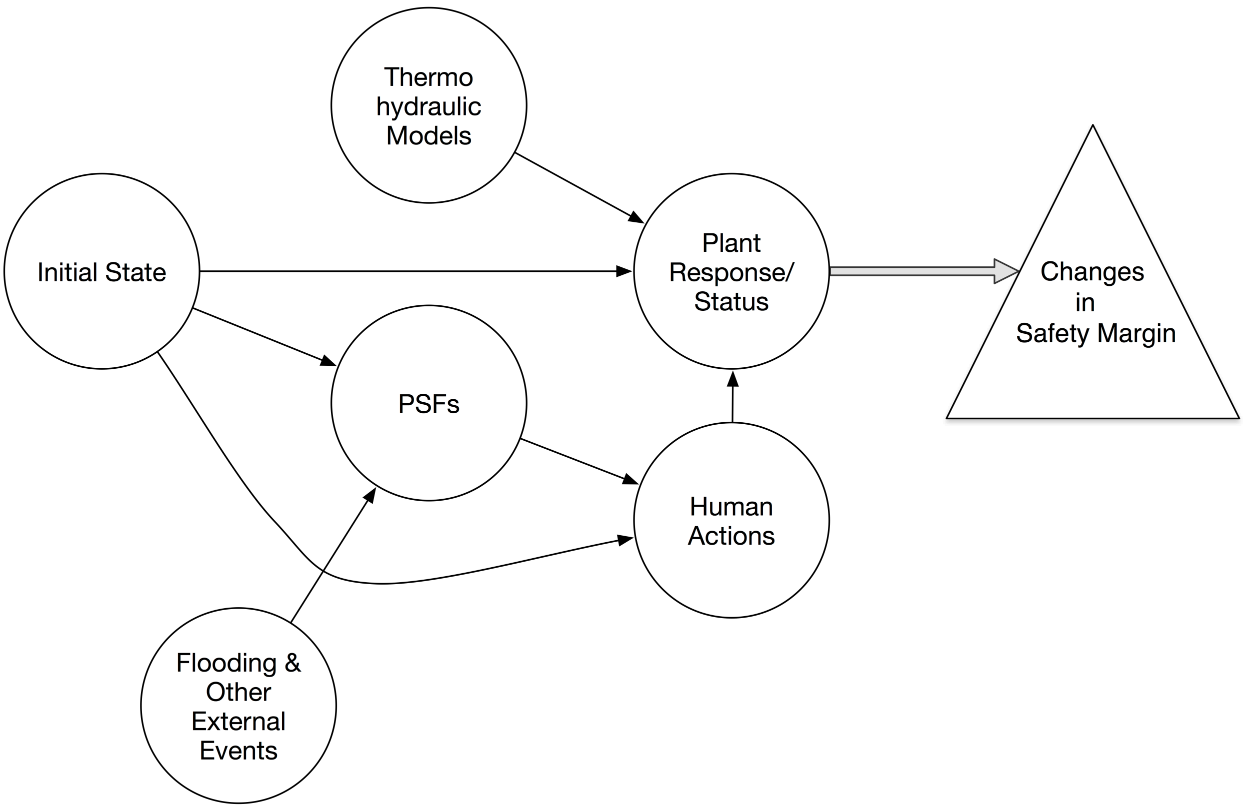


Figure 16: Influence diagram of HUNTER

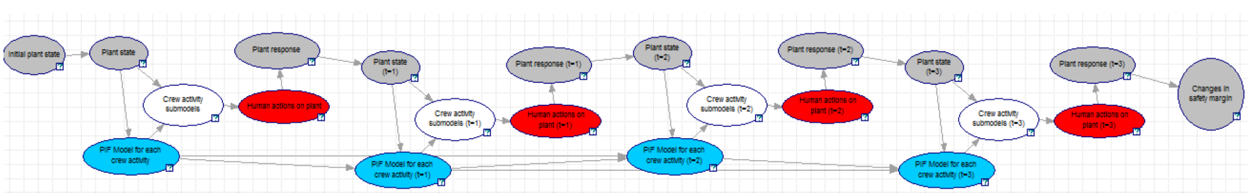


Figure 17: Time sequence view of the HUNTER framework as a Dynamic Bayesian Network

### 4.4.2 Human Actions

Since the focus of computation-based HRA centers on calculating the timing of control actions rather than the probability of a pre-defined HFE, two primary activities need to be conducted:

- Defining a basic unit of analysis (i.e., defining plant control actions), and
- Defining an approach to assessing the timing of those actions.

### 4.4.3 Plant Control Actions

One approach for defining the basic unit of analysis is to develop a standard language and syntax for which actions are performed on the plant. This can be accomplished through predefined *plant control actions* (PCAs) and corresponding *plant interface elements* (PIEs) as depicted in Table 3. The PCAs are those actions the operators (or virtual operators in the case of HUNTER) can perform, while the PIEs are the affected systems on which those actions are performed. This syntax, in symbolic format, becomes the method for communicating between HUNTER and RAVEN.

Table 3: Standard language and syntax for plant control actions

| Plant Control Action (PCA) | Plant Interface Element (PIE) |
|---|---|
| - Change state of binary/discrete component<br>(e.g., Open/Close)<br>- Change state of continuous component<br>(e.g., control, align, adjust, maintain)<br>- Maintain (regulate)<br>… | - Component (e.g., Valve, Pump, PORV)<br>- Subsystem (e.g., -SFPC heat exchanger)<br>- System level (e.g., Pressurizer, SG)<br>… |

Each possible pairing of PCA and PIE would be associated with a probability distribution for the action time. For many events, the distribution will likely be a lognormal distribution. The parameters of the lognormal distributions for each PCA-PIE pair would be contained in a database of those items. This database should be populated with information from existing operating experience, new crew simulator experiments (e.g., for control room actions), and field data collection (e.g., for non-control room actions).

Given the widely-discussed scarcity of data in HRA, the development of these distributions will likely require a Bayesian approach capable of combining expert judgments and available data to provide a robust basis for assigning the parameters of the lognormal (or other selected) distributions for action time. In implementing this Bayesian approach, the prior parameters of the distributions would be assigned by experts, based on rigorous elicitation. The second aspect of this approach would be to update this prior information with data from available HRA data collection activities. These data entail a wide range of international activities comprised of both simulator experiments, operating experience collection, and field data collection (CSNI, 2012; Skjerve & Bye, 2010; Broberg, Hildebrandt & Nowell, 2010; Park & Jung, 2007; Chang & Lois, 2012). The information from existing operating experience and new crew simulator experiments would be used to Bayesian update the distributions for specific activities (e.g., for control room actions). Information from field data collection would be used to update non-control room actions. The final posterior distributions formed by the Bayesian updating process would be used as the baseline values for the distributions, i.e., they would be encoded in the database of PCA-PIE pairs. Specific pairs in this database would be updated annually as new data becomes available.

### 4.4.4 Crew Activity Submodels

Crew activity submodels represent the continuum of cognitive and physical activities involved in formulating a PCA. In this module the crew activities are decomposed into three classes of cognitive activities:

- Information processing (I),
- Diagnosis/Decision making (D), and
- Action Taking (A).

This classification mirrors a long-standing and widely accepted framework in cognition that corresponds to perception, cognition, and behavior. We have adopted the nomenclature found in two recent HRA methods, the Information Decision and Action in Crew Context (IDAC) method (Chang & Mosleh, 2007a) and the Integrated Decision-Tree Human Event Analysis System (IDHEAS) method (Whaley et al., 2012). Information processing represents the input phase associated with sensation and perception by the operators. It is at this stage the operators detect critical information such as plant parameters, alarms, and procedure steps. Decision making occurs when the operator combines the perceived information with knowledge and problem solving. The procedures act as external tools to assist this process. Finally, the operators take action based on the decision they have reached.

Each of these response phases is further divided into a classification of crew activities, i.e., *crew activity primitives*. These crew activity primitives (CAPs) are the fundamental building blocks of human activities in a nuclear power plant. They provide a standard language and syntax for expressing the activities to be completed. In essence, the CAPs would be a complete characterization of:

- Types of information to gather,
- Decisions the operators must make, and
- Human actions that are performed (excluding any plant control actions) in order to reach the PCA.

These building blocks would be used to conduct HRA activities on both proceduralized and non-proceduralized actions, thereby establishing a common approach and language for conducting HRA for both Level 1 and Level 2 or 3 PRA.

Each CAP would have two elements: a noun and a verb. Each CAP would be associated with a model of the interdependent PSFs that affect that specific CAP (see next section). This model would include interdependency among the PSFs, established based on psychological literature. The model would assign an HEP for each CAP, based on the states of the relevant PSFs

The research approach to developing the CAPs is likely to entail a thorough task analysis of current procedures. Each procedural step would be identified as an I, D, or A step. One example of a classification of crew activities is found in the PHOENIX method (Ekanem & Mosleh, 2014) and repeated in Table 4. Note that only the verbs for I and A are derived from PHOENIX.

Table 4: Crew activity primitives from the PHOENIX HRA method

| Information Primitives | | Diagnosis/Decision making primitives | | Action Taking primitives | |
|---|---|---|---|---|---|
| Verb | Noun | Verb | Noun | Verb | Noun |
| - Monitor<br>- Identify<br>- Check /Verify<br>- Collect<br>- Evaluate /Interpret<br>- Record<br>- Compare<br>- Scan<br>… | - Binary parameter state<br>- Discrete parameter state<br>- Continuous parameter state<br>- Threshold comparison<br>- Trend statement<br>- Panel<br>- Procedure<br>... | - Calculate<br>- Select which system to use<br>… | - Value (e.g., required temperature)<br>- System A<br>… | - Coordinate<br>- Notify<br>- Dispatch<br>- Go to | - Person (e.g., RO, senior reactor operator, shift technical advisor)<br>- Co-located team (e.g., multiple persons in the control room)<br>-Area of room or plant |

### 4.4.5 PSF Models for Crew Activities

One goal of the HUNTER approach is to provide a scalable approach with a simplified subset of PSFs to allow proof of concept. HRA methods such as A Technique for Human Error Analysis (ATHEANA; NRC, 2000) contain up to 60 PSFs. IDAC (Chang & Mosleh, 2007a; Coyne, 2009; Li, 2013) contains a similar number of PSFs in a dynamic context, and to date, only a small number of these PSFs have been functionally modeled and implemented. Instead of attempting to model all possible PSFs, it is more feasible to pit a higher level classification such as found in the SPAR-H method or in recent work by Groth et al. (2014). Using the eight PSFs in SPAR-H, for example, allows us to develop a method sensitive to the effects of PSFs without creating overly burdensome degrees of differentiation between observable aspects of performance.

The PSF models would be used to modify the baseline distributions for human action times (which were discussed in Section 4.4.3). The PSFs will provide empirically based distribution tables to show the range of effects possible for the PSF and will provide a first approximation of the effects of the PSFs on operator performance in a dynamic context. The PSF models would contain information about which PSFs are relevant to a specific type of performance as well as information about how the PSFs impact the performance time. The HUNTER concept entails a Bayesian Network (BN) model containing both of these types of information. The relevant PSFs are encoded in the structure of the model. The quantitative effect of the PSF is encoded in the probability distributions associated with the model.

The structure of BN models could be developed via several approaches: one model for each CAP (which is conceptually similar to context-based approaches such as ATHEANA), or one "macro-model" which is relevant for all CAPs (such as in PHOENIX). The BN model would include as many PSFs as necessary to capture the causal paths that lead to failure, whereby irrelevant PSFs would be excluded.

The quantitative relationships in a BN are defined by this structure. For most PSFs, simply encoded logical rules can be used to define the state of the PSFs, as is done in many current HRA approaches (e.g., if indicators are unclear, then the Ergonomics PSF is assigned as "Poor"). The quantitative effect of the PSF on the CAPs would be defined by a combination of expert elicitation and available data developed using a Bayesian updating process similar to the one described in Section 4.4.3. The resulting model would be used to produce a context-specific modification to the parameters of the baseline distribution of action times.

The dynamic PRA simulation controller would assign states to the PSFs based on a combination of direct assignments starting with the plant conditions and some random assignments for aspects of performance that cannot be simulated. These states would be propagated through the BN model, and the resulting probability would be used to adjust the parameters of the action times.

## 4.5 Summary

In this section, we have proposed HUNTER as a computation-based HRA approach to interface with RAVEN as the computational engine. We have outlined several of the requirements for HUNTER and included details toward implementation. To date, HUNTER remains a conceptual approach. It is planned in the next phase of this research project to start implementing and testing HUNTER. Since HUNTER is still very much in its infancy, the details will surely change and evolve over time. However, as envisioned, HUNTER does not reinvent HRA methods. It is clear that HUNTER will make use of existing HRA approaches from both static and dynamic HRA. The unique aspect of HUNTER is that it is a modularized hybrid approach that allows different methods to be interfaced with RAVEN and tested for their utility. It is entirely conceivable that there will be more than a single implementation of HUNTER. As such, it should be considered a multi-method approach for HRA. Much of HUNTER's ultimate implementation (or perhaps implementations) will result from the fundamental research that still needs to be conducted to formulate solid models of operator performance. Those research needs are summarized in the next chapter. These research needs focus more broadly on the research needed for computation-based HRA. Specific development efforts for HUNTER will be articulated in a future report.

# 5. RESEARCH NEEDS

## 5.1 Introduction

In this chapter, we will explore three aspects of computation-based HRA that require research:

- Simplified cognitive modeling,
- Dynamic event modeling, and
- Empirical data collection.

These activities form the basis for understanding operator performance in a dynamic context. While these activities are foundational to building the HUNTER approach, but they are also generic research needs that are important to the broad HRA community. Note that the following discussions are not framed strictly in terms of research needs. Instead, we adopt an approach of exploring the topic within which the research must be conducted. This background information is necessary to understand the current state of knowledge and gaps that require research to address.

## 5.2 Simplified Cognitive Modeling

### 5.2.1 Background

Attempts to model the entirety of human cognition are folly. As noted in Deutsch (2012), despite roughly six decades of artificial intelligence, there is as yet no computational demonstration even closely approximating human cognition. The focus is too broad; the processes, too poorly understood; and the complexity, too great. Artificial intelligence systems still lack the ability to mimic how the human brain creates explanations. Without this fundamental skill, they are simply stores of rules of decision making or databases of knowledge.

There are multiple ways to frame cognition for the purposes of artificially modeling human intelligence. In the levels of explanation approach (a.k.a., the Tri-Level Hypothesis), it has been postulated that each cognitive subdiscipline answers a different type of question (Marr, 1982; Pylyshyn, 1984; Anderson, 1993; Dawson, 1998). Broadly speaking, there are *computational*, *procedural*, and *physical* levels of explanation, answering *how*, *why*, and *what* cognition occurs, respectively. Artificial intelligence assumes that human cognition has a computational basis. It follows that since cognition is computational, it may be computationally modeled. This computational approach makes artificial intelligence quite different from the other subdisciplines of cognitive science, because it puts much more an emphasis on how cognition occurs than on what, when, where, or why. Psychology, for example, has concentrated on the identification of cognitive facts (*What is cognition?*) or on the reaction times of cognitive processes (*When does cognition occur?*); linguistics has focused on the categorization of language (*What role does language play in cognition?*); neuroscience has specialized largely in the localization of cognitive functioning (*Where does cognition occur?*); and, philosophy has entertained the broader questions of cognition (*Why does cognition occur?*) as well as the foundational and definitional questions of cognitive science (*What is the mind? What is consciousness?*).

Of course, we are painting a picture of the cognitive subdisciplines with a very wide brush. Generalizations aside, the point to be made is that artificial intelligence has a unique set of objectives and methods in cognitive science. Artificial intelligence seeks primarily to understand cognition to the degree that it may be reimplemented or adapted for computer use (*How does cognition occur?*). The psychological, linguistic, neuroscientific, and philosophical approaches have complemented and even facilitated artificial intelligence, but they have not been the primary sources of artificial intelligence theory (Johnson-Laird, 1993).

According to functionalist philosophy, the underlying biology of the brain is not the ultimate determiner of cognitive functioning (Churchland, 1988). Artificial intelligence presents cognitive science with its most functionalist stance. The premise of artificial intelligence can be summarized simply as follows: if cognition is not solely realizable in a human neurological system, then cognition may be simulated via another medium, specifically via computer modeling. The field of artificial intelligence posits that natural cognition is no different than artificial cognition. Thus, the functions of the organic, neural, analog, parallel-functioning human brain can be duplicated by an inorganic, symbolic, digital, serial computer system.

The functionalist account of human cognition is not without criticism (Block, 1993), but it remains an underlying assumption of most artificial intelligence research. Clearly, it needs to be a tenet of artificial intelligence that cognition is not limited to literal human biology. Otherwise, there would scarcely be any point in attempting to simulate human cognition using anything less than a human brain. There are, however, those semi-functionalist artificial intelligence researchers who argue for a more humanlike computational architecture in order better to simulate human cognition. These researchers, who most notably include the connectionists (Macdonald, 1995; Fodor & Pylyshyn, 1988), stipulate networks of neurons, like those in the human brain, in the hope that such a system will yield humanlike cognition. Connectionist systems have indeed proven interesting in their realistic replication of some aspects of human learning, including language processing, but they have not significantly challenged the more traditional functionalist artificial intelligence framework (see the interview with Allen Newell, pp. 145-55, in Baumgartner & Payr, 1995).

It is important not to forget that the functionalist position holds, to a greater or lesser extent, for both classical and connectionist artificial intelligence systems. Classical artificial intelligence, including the variety commonly utilized in production systems, takes the extreme functionalist stance that the underlying structure of the information processing system need not be similar to the human brain in order to simulate its functioning. Connectionist artificial intelligence, such as the parallel distributed processing architecture introduced by Rumelhart and McClelland (1986), takes a much milder stance. It aims to mimic the structure and functioning of the human brain through simulated neural networks. Whereas a classical artificial intelligence system is based largely on the serial computer as a metaphor for information processing, connectionist artificial intelligence simulates–to a degree–parallel neurological processing. There is ongoing debate between classical and connectionist artificial intelligence camps regarding which paradigm provides the clearest picture of how the mind works (Garson, 1994). Despite these often hefty debates, both approaches to artificial intelligence share the functionalist position that human cognition can be simulated using some architecture other than the human mind. It is the details of the simulation that are contested. Classical artificial intelligence does not require a high degree of biological fidelity in its simulations; connectionist artificial intelligence attempts a higher fidelity.

The philosophical debate over functionalism is further complicated by the issue of consciousness, that omnipresent yet empirically elusive component of humanity. The question of consciousness is whether it is a fundamental component of cognition. If it is, it arguably needs to be included in any complete model of human cognition and it needs to be considered in greater detail by artificial intelligence researchers. To this end, there are weak and strong models of artificial intelligence. The weak variety makes no claims that an artificially intelligent system could (or should) be conscious; the strong variety claims that it would be possible for such a system to be conscious. The topic of consciousness has become a very highly debated one, one closely associated with the materialist and dualist perspectives long carried out in philosophical and scientific communities. As the arguments regarding consciousness have become more heated in recent years, artificial intelligence has increasingly entered the spotlight. Proponents of functionalism will claim that advances in artificial intelligence have proven that cognition—including consciousness—are not determined by the form of the mind (Flanagan, 1992) and should be simulatable by computers. In contrast, anti-functionalists take issue with the fact that consciousness could ever be cast outside of the context of human cognition in the human brain. There are also those who hedge either side of the argument, like Chalmers (1996), who claims that consciousness is part of human cognition but represents an as yet unclassified component. In Chalmers' view, it should be possible to create artificial consciousness, although only after the true functioning of natural consciousness is understood.

Table 5: Four types of artificial intelligence (after Russell and Norvig, 1995).

| | **Intelligence** | |
|---|---|---|
| **Behavior** | Systems that think like humans. | Systems that think rationally. |
| | Systems that think and act like humans. | Systems that think and act rationally. |

There are four basic goals of artificial intelligence (Russell & Norvig, 1995). As seen in Table 5, these goals are comprised of the method of intelligence being implemented (humanlike or rational) and the behavior of the artificial intelligence system (thought only or thought and action). The rational system functions according to syllogisms—premises and consequent conclusions—to approximate limited cognitive skills. The humanlike system is more ambitious in that its intelligence must incorporate and mimic the entire realm of human cognitive skills. Its aim is to be able to pass the Turing test, meaning the simulated cognition must be sufficiently plausible to fool a human investigator into believing the cognition originated in a human (Turing, 1950; Hunt, 1993). The artificial intelligence methods most closely associated with HRA are those that attempt to mimic human thoughts and act like humans (see the lower left quadrant of Table 5).

Within the artificial intelligence community in general, little emphasis has been placed on attempting to create a comprehensive repertoire of humanlike thoughts and actions. Humanlike actions include overt signs of cognition such as language utterances and bodily movements. The field of natural language processing continues its efforts to simulate human speech comprehension and production in computer systems, but there are many technological and conceptual bottlenecks that have slowed the progress therein (Russell & Norvig, 1995). Bodily movements probably have more to do with motor functioning than with cognitive abilities and have been the research specialty more of robotics, biomechanics, or ergonomics than of artificial intelligence, although the work of Brooks (1997) does successfully bridge robotics and cognition.

Most artificial intelligence work on humanlike action concerns empowering computer systems with the ability to act on decisions in a manner similar to humans. Such action entails producing a change in the environment, but there is a much greater concentration on the product of the environmental changes rather than the means toward these environmental changes. For example, if an artificial intelligence system can open a door electronically, the product is an open door. This product is the same as when a human physically walks across the room, turns a door handle, and pulls the door toward him- or herself. Artificial intelligence is not always concerned with the means of opening the door, as long as the computer-generated goal to open the door has the same overall outcome as the human-originating goal to open the door.

Simulating humanlike action goes beyond the merely computational role of artificial intelligence. In general, artificial intelligence has no central interest in what processes went into opening the door, what the language is by which the door is represented, where the idea of opening the door originated in the system or human, or why the computer or human might want to open the door. Artificial intelligence has as its goal looking at how the computer and the human generate the thought to open the door. In many ways, this is the opposite of behaviorism, the once dominant research field of psychology. Whereas behaviorism is primarily concerned with the overt actions (or behaviors) of the organism, artificial intelligence is primarily concerned with the thoughts behind the actions.

### 5.2.2 Path Forward

This rich framework of artificial intelligence is self-limited by the enormity of the undertaking. Without constraints on the endeavor, RISMC and HUNTER stand no greater assurance of success than the preceding decades-old research efforts in artificial intelligence. How can computation-based HRA therefore hope to succeed where other efforts are still not successful? There are three areas of HRA for nuclear power operations that make it a unique testbed:

- *PSFs*: Most artificial intelligence research is stripped of internal and external influences on cognition. Yet, a pure reasoning engine is not an accurate reflection of human cognition, because human cognition never occurs in a vat. Instead, human cognition must process complex environments, competing and conflicting goals, and even demanding distractions (such as emotions) from pure reasoning. Humans are not purely logical

Vulcans as depicted in the *Star Trek* television and film series. Humans must balance rationality with emotions and other factors that affect cognitive processes. The PSFs used and understood in HRA provide tools to inform cognitive processes in realistic contexts. PSFs like stress directly impinge on the quality of decision making. While PSFs can complicate the decision making algorithms, PSFs also provide a finite set of factors to consider in modeling cognition. PSFs can serve to reasonably constrain the extraneous aspects of modeling operator cognition. PSFs afford researchers a finite number of considerations within the broader, infinite domain of human cognition.

- *Procedures:* Likely no field outside nuclear energy contains a more regimented set of operator actions and decision points. Plant operating procedures provide the logical starting point for guiding operator decisions and actions, especially within design-basis activities. As previously noted, not all operator actions are a direct reflection of the procedural script, but the procedures provide a solid initial outline for the range of logical operator actions.
- *Controlled Environment:* There are billions of contexts and situations in which the fully cognizant human finds him- or herself. A full representation of human cognition must encompass this limitlessness. An advantage of the control room in which operators function is that it is finite. Of course, the plant is not a simple system, and there are countless permutations of plant states that operators must confront. Still, the problem space is constrained in a reasonable manner, allowing researchers to focus on a narrow range of behaviors and decisions as is appropriate for operating the plant. Aspects of human cognition beyond the control room are simply beyond the scope of HUNTER.

There is likely little value to HRA in engaging in the philosophical discussions and debates on artificial intelligence as presented earlier in this section. The true value of artificial intelligence and human performance modeling comes from harvesting relevant models of cognition that can guide operator performance. Of particular interest are those simplified models that can help account for why a person would decide on Action A vs. Action B. Basic rules for decision making, including the influence of PSFs on that decision making, are missing from current HRA. Research for computation-based HRA needs to focus on extracting and simplifying existing models of decision making available in the artificial intelligence literature.

## 5.3 Dynamic Event Modeling

### 5.3.1 Background

As noted in Boring (2014), there exists no single or standard way to decompose human activities into an HFE. In practice, the HFE is defined as the entirety of human actions related to the human interaction with a particular system. In other words, the HFE is defined top-down, from the PRA level of interest, to encompass all human actions that can contribute to the fault of a component or system modeled in the PRA. In other domains, where such top-down HFEs are not clearly prescribed, the HFE may be built bottom-up, starting with human actions and clustering them as they interact with a component or system. The bottom-up approach is conducted by human factors analysts who will typically follow a task analysis approach to building the HFE (Boring, 2015). The issue centers on the possibility that the two approaches may not always converge on the same HFE. How many and which actions are clustered into an HFE is unclear in the two approaches.

HRA has created tools to help address the boundaries between HFEs. Most HRA methods consider *dependence*, which is the relationship between HFEs. A common assumption in HRA methods is that error begets error, meaning an initial human error tends to prime subsequent errors, increasing their likelihood. As elaborated in Whaley et al. (2012), it requires a significant break in the evolution of the event that results in a changed crew mindset to disrupt dependence or recover from the error. If the crew does not realize that an error has occurred, they will tend to continue actions based on false assumptions, thus propagating the initial error. Mathematically, dependence is commonly treated such that it results in an increased HEP on subsequent HFEs. A correction factor is applied to the calculated HEP for the HFE to increase that number. The higher the dependence between two HFEs, the higher the likelihood of error on the second or subsequent HFE.

The preceding discussion has centered on HFEs and dependence for conventional HRA, which is static in nature. Once the overall system is modeled, including HFEs, they do not change as a result of the event progression. Dynamic or computation-based HRA does not rely on a fixed set of event and fault trees to model event outcome. Rather, it builds the event progression dynamically, as a result of ongoing actions (Acosta & Siu, 1993). The dynamic approach in PRA has proved especially useful for modeling beyond design basis accidents, where not all failure combinations (and, importantly, not all recovery opportunities) can be anticipated or have been included in the static model. Additionally, the failure of multiple components or unusual sequences of faults, even within design basis, may challenge the fidelity of the PRA model. While such events are rare, dynamic modeling affords the opportunity to anticipate such permutations and address them in a risk-informed manner should they occur.

Most of dependence as used in HRA is based on the dependence model in the original HRA method, the Technique for Human Error Rate Prediction (THERP) found in NUREG/CR-1278 (Swain & Guttman, 1983). The key guidance for this approach is found in Chapter 10 of NUREG/CR-1278. The key types of dependence discussed in THERP are found in Figure 18. To illustrate, assume two tasks occur sequentially, first Task A and then followed by Task B. *Independence* means that the success or failure in Task A has no bearing on the success or failure of Task B. In contrast, *dependence* occurs when the success or failure of Task A does influence the success or failure of Task B. *Direct dependence* means that Task A expressly influences Task B. These are typically closely coupled tasks, where the outcome of the first necessarily affects the second task. In contrast, *indirect dependence* occurs when both tasks share a common mediating influence such as a mutual PSF. Swain and Guttman suggest stress is such a PSF, whereby an operator experiencing high stress will see deleterious effects on all tasking he or she performs. The PSF in this case acts as a type of common cause leading to elevated error rates for both tasks. For direct and indirect dependence, there is both negative and positive dependence. *Negative dependence* implies an inverse relationship between the two tasks, e.g., success on Task A increases failure (decreases success) on Task B or failure on Task A increases success (decreases failure) on Task B. *Positive dependence* implies a positive relationship between two tasks, e.g., success on Task A increases the chance of success on Task B or failure on Task A increases failure on Task B.

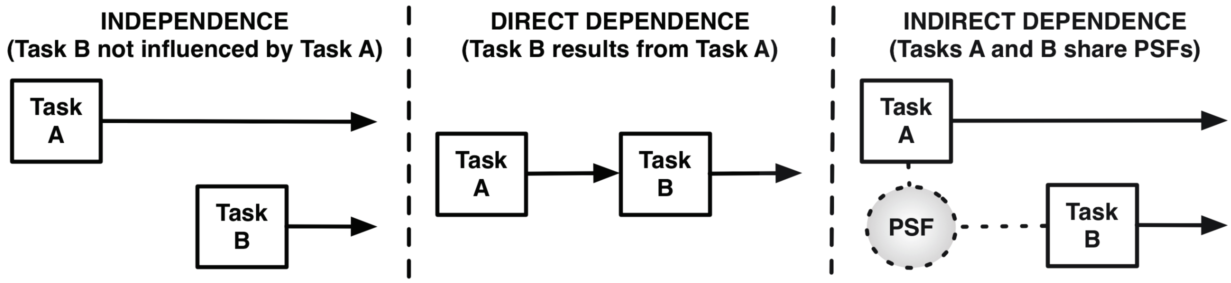


Figure 18: Three types of dependence considered in NUREG/CR-1278

Because actual performance data are often scarce and because estimating dependence without calibration to a scale is highly subjective, THERP provides the Positive Dependence Model. In this approach, a mathematical correction is applied according to the level of dependence. Dependence is assumed at five stations along a continuum, ranging through zero, low, moderate, high, and complete dependence. Determination of the appropriate level of dependence is guided in Table 10-1 in THERP. The correction factors, found in Table 10-2 in THERP, range from no change over the basic HEP for the task if zero dependence up to an HEP = 1.0 for complete dependence, the likelihood of error increasing the greater the dependence. Similar corrections are applied if considering task success, with the likelihood of success increasing the greater the dependence between two tasks. In practice, HRA rarely considers success space, and the predominant use of dependence focuses on failures and HEPs.

THERP's Positive Dependence Model remains the dominant approach to calculating dependence in HRA and is featured in most contemporary HRA methods (Kolaczkowski et al., 2005). For example, the SPAR-H method (Gertman et al., 2005) adopts the same levels of dependence and correction factor calculations as the original THERP method. While the Positive Dependence Model is widely deployed, it is often used slightly differently than in the original implementation. In THERP, dependence was historically calculated between subtasks, not between HFEs.

Subtasks are modeled in the HRA Event Tree, which is unique in THERP (see Figure 19). It has in practice been replaced by event and fault tree logic aligned with PRA modeling conventions. THERP's HRA Event Tree is not synonymous with these approaches, and THERP's mathematical approach to joining subtasks can be lost in translation. NUREG/CR-1278 Chapter 5 particularly notes that in fault tree representations dependence is much more difficult to represent compared to the equivalent HRA Event Tree representations. The HRA Event Tree guides the calculation of the total HEP for the HFE. The probabilities of subtasks along each failure path are multiplied, and these subtask probabilities are then summed. In the process of multiplying the subtask probabilities, the correction factor for dependence is applied where appropriate. Because THERP provides lookup tables for subtask HEPs, the proper level of analysis granularity is ensured.

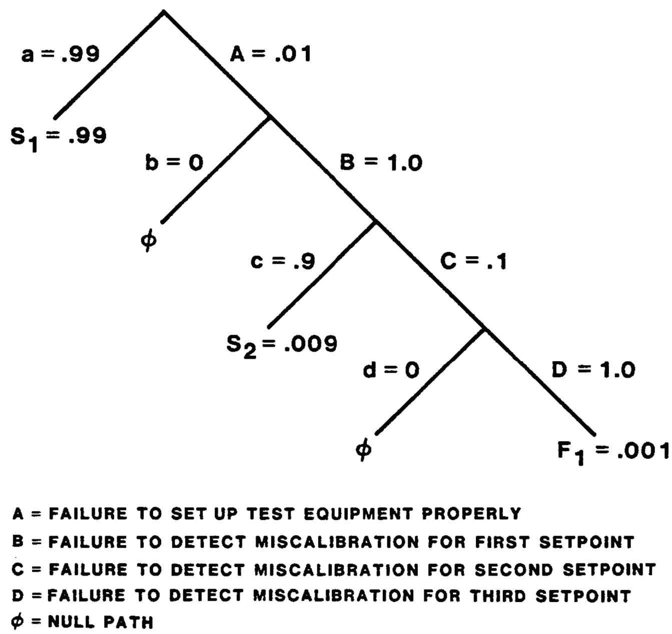


Figure 19: Example THERP HFE comprised of subtasks within an HRA Event Tree (from NUREG/CR-1278, Figure 10-1)

Of particular importance is the current practice of applying the Positive Dependence Model *between* HFEs. THERP originally considered dependence *within* HFEs only. In fact, in our interpretation, the boundary between HFEs might be considered the point at which there is no logical dependence between subtasks. In other words, the very definition of an HFE might be the case of clustering dependent subtasks, while independent subtasks form the boundaries *between* HFEs. Thus, using the Positive Dependence Model between HFEs may violate key assumptions about the nature of subtasks and HFEs. Please note that I do not wish to claim that the current practice of applying dependence between HFEs is wrong nor that it produces invalid HEPs; rather, I am simply pointing out that current practice does not appear to follow the original intent of the Positive Dependence Model.

It should be noted that an alternative approach to the standard Positive Dependence Model is provided in Appendix B of NUREG/CR-1278 and credited to Easterling (1983). The Positive Dependence Model effectively models direct dependence. Although it may also be applied to indirect dependence, it remains insensitive to the effects of any mitigating or mediating PSFs. Appendix B of THERP provides a probabilistic treatment of indirect dependence. The equations provided account for the influences of PSFs in addition to the simple coexistence of Task A and Task B, whereby the conditional probability of Task B given Task A and the PSF influence can be calculated. As in the Positive Dependence Model, a greater level of dependence between the three factors results in a higher HEP.

One challenge of dynamic HRA is that the unit of analysis is not necessarily the HFE. Dynamic HRA represents a continuous evolution of the event, including multiple discrete actions. The problem of determining the HFE is therefore analogous to the bottom-up approach for defining HFEs, based at the task level. Each subtask within the HFE carries with it properties that affect the probability. In fact, it should be possible to calculate the HEP at any point in time for the activities currently exercised by the human operator. This derivative HEP is not for the entire HFE, but rather for a discrete moment in time. Yet, the combinatorial aspects of these HEPs within the HFE are not expounded in existing HRA methods. Ideally, the integral of the dynamic HEPs should equal the static HEP for the HFE. This bridge between static and dynamic HEPs has not been established to date and presents a challenge when applying dynamic HRA methods to existing HRA problems. Without a clear definition of the unit of analysis (i.e., the HFE), it is impossible to quantify the error likelihood.

The key to linking the subtasks in dynamic HRA to an overarching HFE umbrella is to use task dependence. However, the existing approach for dependence in HRA falls short of providing a method that could function for dynamic HRA needs.

### 5.3.2 Path Forward

Topics for future research to help realize dynamic HRA and, eventually, dynamic dependence include:

- Defining HFEs dynamically, such that they make use of bottom-up approaches and can emerge as part of the dynamic progression of the event rather than rely on predefined characterizations of human activities.
- Automated determination of dependence levels, such that correction factors for dependence can be applied as part of the dynamic HRA modeling process without the need for subjective level assessments by human analysts.
- Articulation of a mathematical conditional probability formula, building on the discussion on indirect dependence by Easterling (1983) in THERP and likely incorporating contemporary methods for Bayesian conditional probabilities to account for the influence of previous subtasks and PSFs.
- Validation of the mathematical treatment of dependence, including review of the dependence correction factors included in THERP's Positive Dependence Model and their applicability to dynamic calculations.
- Modeling of PSF distributions to account for the variable influence of the PSFs over time on operator performance, specifically to account for PSF lag and linger.
- Modeling of PSF overlaps to determine the extent of error spilling in simple to complex events.

Dynamic dependence is an essential part of using dynamic HRA to compute HEPs. Future research within RISMC will aim to create an implementation of dependence that will serve the needs of dynamic HRA modeling while improving and validating the dependence approach used in static HRA. HRA's approach to dependence has remained largely unchanged since THERP, the first HRA method. Yet, paradoxically, dependence is not used in practice in the subtask manner originally intended in THERP. Dynamic dependence requires ongoing subtask analysis, suggesting the importance of revisiting the THERP subtask dependence approach. At the same time, it is crucial not only to revisit past approaches but also to include systematic research on developing new approaches to dependence as needed. It is now time to reconsider how dependence is treated in HRA.

## 5.4 Empirical Data Collection[b]

### 5.4.1 Background

One of the persistent challenges of HRA remains how to obtain enough data to understand human performance. Especially given HRA's strong foundations in nuclear energy, the opportunity to build up an extensive corpus of human performance data is difficult given the complexity and cost of studies involving nuclear power plant operators. Database efforts such as Nuclear Computerized Library for Assessing Reactor Reliability (NUCLARR; Gertman et al., 1990), Human Event Repository and Analysis (HERA; Hallbert et al., 2006), and Scenario Authoring, Characterization, and Debriefing Application (SACADA; Chang et al., 2014) have attempted to look at nuclear power analyses and events as a means of expanding the data basis in HRA. This approach has thus far proved inadequate to significantly expand the data underlying HRA. In the case of NUCLARR, event analyses were built on other event analyses, and the database became too circular to prove a robust data source. Work on HERA and its successor, SACADA, is ongoing, but the events analysed in HERA and SACADA are too infrequent to provide quantitative insights in the form of human error probabilities. Chang et al. (2014) have devised a method to elicit data from training scenarios in control room simulators, which may prove an effective way to gain human performance data below the threshold of reportable events. Tran et al. (2007) and Griffith and Mahadevan (2011) have suggested the use of meta-analytic techniques to generalize data. This approach was incorporated into the National Aeronautics and Space Administration (NASA) HRA database (Boring et al., 2006), which features a taxonomy that allows HRA insights to be gleaned from multiple sources, including traditional human factors studies outside the aerospace or nuclear domains. A similar approach for generalizing results is used in the CORE-DATA (Gibson, Basra, & Kirwan, 1999), in which human performance insights are extracted from multiple domains. The CORE-DATA serves as the underlying data basis for the Nuclear Action Reliability Assessment (NARA) and Controller Action Reliability Assessment (CARA) HRA methods, among others still in development. Generalizing data from different sources to nuclear power operations is also the key idea in Jing, Lois, and James (2011), which provides a preliminary framework for mapping human errors across domains.

The research framework presented in this section builds on prior efforts to create a solid data basis for HRA by investigating the opportunity for data collection from multiple sources. Table 6 lists the types of experimental environments currently underway. The aim of this research is to bridge different capabilities at several research facilities, each with a shared goal to produce insights into human performance that can inform HRA. Following is a brief discussion of each research area prior to a discussion of the synthesis of the separate projects in support of RISMC.

[b] This section is based on a paper by Boring, Kelly, Smidts, Mosleh, and Dyre (2012).

Table 6: Crosswalk of different empirical research platforms for HRA.

| Title | Microworld | Generic Simulator | Training Simulator | Simulation |
|---|---|---|---|---|
| **Location** | University of Idaho | Ohio State University | Idaho National Laboratory | University of Maryland/University of California at Los Angeles |
| **Apparatus** | Low fidelity simulator of process control | Full-scope boiling water reactor simulator | Full-scope pressurized water reactor simulator | Virtual operation and operator model |
| **Participants** | University students with minimal training | University students with formal training | Licensed operators | Virtual operators |
| **Interface** | Simplified graphical user interface | Enhanced plant interface with advanced overview displays | Digital mimics of analog instrumentation and controls | Interface to plant model and thermohydraulics through software |
| **Metrics** | Primarily performance data (e.g., reaction times and accuracy) automatically logged through experimental trials | Observational performance logging coupled to simulator logs; crew debriefs | Observational performance logging coupled to simulator logs; crew debriefs; physiological measures; eye tracking | Performance over repeated trials, typically in the form of success/failure metrics and timing |
| **Advantages** | Ability to run large number of experiments suitable for inferential statistics and first principles research | Authentic plant model; ability to run large number of student operators; flexibility to incorporate new interface elements | Authentic plant model identical to one used for operator training | Ability to run unlimited simulations within the constraints of what can be modelled in the system |
| **Limitations** | Limited fidelity of simulator; limited expertise of student participants; constrained ability to study team collaboration due to solo participant interface within microworld | Some limitations of student crew training and experience—it is unrealistic to test student operator performance on certain complex scenarios for which they do not have same versatility as licensed crews | Limited availability of licenced operators for experimental work; because of complexity of plant control, difficult to control for confounds in operations such as multiple faults resulting from a faulted system | The fidelity of the simulation is limited by the quality and completeness of the underlying modeling of the system; considerable development effort to incorporate new features and scenarios |

### 5.4.2 Microworld Simulator

Microworlds are a type of simplified simulator that allows users to learn about the domain being simulated. They differ from simulators, which generally strive for maximum realism. Whereas users must generally be extensively trained to operate simulators, microworlds are often software tools built to help the users understand the concepts and build up an understanding with minimal training. Whereas an electrical grid simulator might offer a realistic model of electrical distribution including an interface that directly mimics the real control system interface that grid controllers would use, a microworld would feature a simplified model and interface designed to help the would-be controller understand the fundamental concepts of grid control. In many cases, microworlds provide the opportunity for training on the interaction between different stakeholders in a complex system environment. This is, however, not a strict requirement of microworlds, and successful microworlds may also feature a single user.

In the context of HRA for NPPs, there is a need for more flexible, varied, and expedient research studies than are possible in a full-scope nuclear power plant simulator. A microworld simulator fulfills this requirement by providing a simplified, readily adaptable model to which non process experts can be trained. Microworlds such as the Dual Reservoir System Simulator (DURESS;

Orchanian et al., 1996) even allow students to be trained to functionally equivalent tasks to those that might be performed in NPPs. The microworld, because it is not tied to a specific plant model, can be customized in a way that real-world plant models cannot while still affording the opportunity to simulate the complex multitasking that is present in real-world control rooms. These results from microworld experiments should not, of course, be generalized back to NPP control rooms without validation.

### 5.4.3 Simulators

Simulator technology for domains such as aviation emerged in the 1930s with the invention of the Link Trainer, a device that allowed pilots in training to learn to manipulate flight controls in a rudimentary manner (Robertson Museum and Science Center, 2000). It was not until considerably later—with advances in computing technology—that mathematical system models and computer generated imagery could be harnessed to create realistic, virtual flight simulations. A similar course was followed for nuclear power plants—initial non-operational hardware mockups of control room panels gave way to entire control room simulators with functional control panels interfaced with underlying thermohydraulic code. Nuclear power plant simulators evolved from being static training representations to interactive, operational systems that could be used to train and test reactor operators' knowledge of plant states and scenarios.

The International Atomic Energy Agency (IAEA, 2004) highlights the historic development of training simulators. Beginning in the 1970s, computerized control room simulators were put in place at centralized facilities to help train control room operators. These simulators were limited by a lack of fidelity in terms of control panel layouts and underlying thermohydraulic code, making them useful for teaching basic plant principles to operators but less useful for plant-specific training. By the 1980s, the fidelity and availability of simulators was greatly increased, and by the 1990s, it became the norm internationally for each plant to have a high-fidelity plant-specific simulator.

The IAEA (2003) defines different types of plant simulators. These include:

- *Basic principles simulator*—which provides a simulation of general concepts relevant to the operation of a plant without providing a faithful mockup of a specific plant.
- *Full-scope simulator*—which is a faithful replica of a specific plant control room and its operations.
- *Other-than-full-scope control room simulator*—which closely mimics a plant but deviates from its human-machine interface.
- *Part-task simulator*—which only models specific systems of a plant.

The term *training simulator* is synonymous with a full-scope simulator as would be found at a nuclear power plant. All simulator types may be used as part of an effective training regime, but there have been increased emphasis on and requirements for training in full-scope simulators. The considerable demand on plant training simulators was already evident in 1992, when a survey suggested that single-reactor site training simulators were used an average of 2000 hours annually across two daily shifts (INPO, 1992). Double and triple reactor sites saw an even greater utilization of their simulator facilities. To counteract the high use of these simulators, control room simulators have been created separate from plants, to serve the primary purpose to conduct research, not to train reactor operators. These are *research simulators*.

While striving to create a realistic plant environment for operators, a research control room simulator provides the opportunity to design and validate new hardware and plant models. New hardware and plant models may prove difficult to implement in training simulators, which are closely tied to the actual plant. This reconfigurable aspect of research simulators affords a unique opportunity to test actual human operators. A significant advance of incorporating human-in-the-loop testing is the ability to estimate the safety of novel control room equipment and configurations. Such a control room simulator serves an emerging research need to collect data on operator performance using new control room technologies. Moreover, it can serve to provide an empirical basis for human reliability modeling used in the certification of plant safety.

A full-scope plant simulator comprises several layers of systems as depicted in Figure 20. At the heart are system models that interact to create a realistic model of plant behavior, including thermohydraulic software modeling using RELAP, a vendor-specific simulator platform (e.g., simulator software development packages by GSE, WSC, and L3), and a plant-specific model executed on the simulator platform. These models combine to form the back end called the engineering simulator. The engineering simulator interfaces with the front-end simulator, which consists of the control room human-system interface (HSI) that the operator uses to understand plant states and control plant functions. The front-end simulator may take many forms such as an analog hard panel system found in typical U.S. training simulators or a digital soft control system found in some foreign plants and research and development simulators. Digital soft control systems may take the form of mimics to analog plant or may represent advanced instrumentation and controls (I&C) that incorporates features such as overview displays and information rich trending displays.

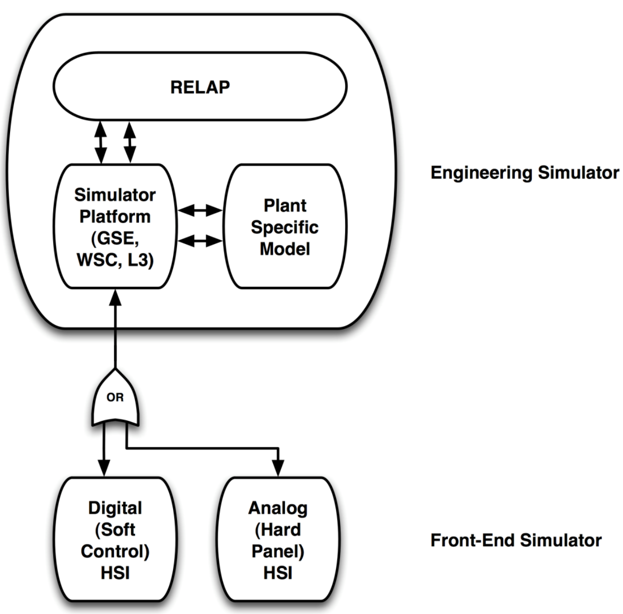


Figure 20: Different components of a plant simulator.

The Humans Systems Simulation Laboratory (see Figure 21) at INL is a platform- and plant-neutral environment intended for full-scope and part-task testing of operator performance in various control room configurations (Boring et al., 2012 and 2013). Currently, plant-specific simulators are coupled to the existing configuration of the plant and are impractical or difficult to reconfigure to test new designs. The INL facility is not limited to a particular plant or even simulator architecture. It currently supports engineering simulator platforms from multiple vendors using digital interfaces. With reconfigurability, it is possible to switch the I&C—not just to digital panels but also to different control modalities such as those using greater plant automation or intelligent alarm filtering. The intent is that licensed plant operators can use the facility as a research simulator, because there is limited availability of the plant training simulator.

Figure 21: The Human Systems Simulation Laboratory at INL.

### 5.4.4 Human Performance Modeling

It is important to note a key distinction here between *simulation* and *simulator* data. Simulations use virtual environments and virtual performers to model the tasks of interest. In contrast, simulators use virtual environments with actual human operators (Bye et al., 2006). In most cases, simulations and simulators may both be used to model dynamic human performance and reliability, as both produce a log of performance over time and tasks. Because simulators use real humans, it is possible to capture the full spectrum of human performance for a given task, whereas simulations must rely on those performance metrics that can be modeled virtually. However, simulations afford the opportunity to perform a wider spectrum of modeling and typically allow easier and more cost effective repeated trials than those tasks involving humans. A large number of trials involving actual humans in simulators is possible but typically requires seeding or forcing an error likely situation in the simulator runs, which may prevent a high level of scenario realism.

Human performance simulation, as outlined in this report, uses virtual scenarios, virtual environments, and virtual humans to mimic the performance of humans in actual scenarios and environments. What sets this form of HRA apart is that it provides a dynamic basis for HRA modeling and quantification. As noted in Section 2.2, traditional HRA methods, by any definition, have featured largely static task analyses of operating events as the underlying basis of performance modeling. These methods have also relied on performance estimations mapped to similar previous performance derived through empirical data or expert opinion. Computation-based HRA differs from its antecedents in that it is a dynamic modeling system that reproduces human decisions and actions as the basis for its performance estimation. Computation-based HRA may use a frequentist approach for calculating HEPs, in which varieties of human behaviors are modeled across a series of Monte Carlo style replications, thus producing an error rate over a denominator of repeated trials. Computatiion-based HRA may also augment previous HRA methods by dynamically computing performance shaping factor levels to arrive at HEPs for any given point in time.

Meister (1999) suggests that HRA filled an important void early in the evolution of human factors by centering on prediction. Much of classic human factors has centered on the collection of data on the interaction of humans with designed systems. The purpose of such data is to improve the design of the system, ultimately to optimize human performance in terms of criteria such as usability, efficiency, or safety. HRA has instead attempted to predict human performance, specifically human errors, that can occur in such human-machine interactions. The purpose of HRA is therefore not typically to improve the design of the system so much as to determine what factors impact the safe human operation of that system. Over time, HRA has been joined by another predictive tool, namely human performance modeling and artificial intelligence. HPM is an umbrella term used to describe systems that simulate human decision making and actions. HPM is largely synonymous with cognitive simulation and artificial intelligence, although it has in practice applied to unified systems that attempt to account for a broad range of human cognitive activities.

#### 5.4.5 Path Forward

The three separate data generation methods discussed in Section 5.4—microworlds, simulators, and simulation—come together in an important way in this project. All three data sources represent different ways of capturing HRA-relevant data (see Table 6):

- Microworlds allow large scale data collection using simplified process control.
- Simulators allow high fidelity data collection using skilled operators.
- Simulation allows generation of virtual performance data based on specified parameters.

To date, the most widely espoused method of collecting data for HRA has been from simulator data. However, the fiscal and practical constraints of running large numbers of crews across large numbers of scenarios suggests this approach will not yield significantly more data than has been generated since HRA's inception. The other three data sources identified in this paper do hold the promise of being more economical and more feasible:

- Microworlds such as DURESS at can be run inexpensively using university students as operators on simplified tasks.
- The research simulator at universities may features a cohort of Nuclear Engineering students who are emerging subject matter and operations experts.

- The ADS/IDAC simulation platform offers literally limitless virtual operator runs to identify potential human performance issues.

The problem is that none of these alternative data sources feature actual operators in actual control rooms. As such, the quality and generalizability of the findings may seem limited in terms of their application to HRA for nuclear power. Tasks are being identified that may prove comparable across the different experimental platforms. The *Boiling Water Reactor and Pressurized Water Reactor Off-Normal Event Descriptions* (NRC, 1987) is being reviewed to find common scenarios that can be run on full-scope research simulators and that can be simplified to run in a microworld environment. Similarly, candidate scenarios will be coded into ADS/IDAC to compare the virtual operator performance against unskilled student operators on microworlds, skilled student operators at university simulators, and licensed commercial operators at INL.

The key to understanding the similarities and differences between the data sources is to benchmark them against each other using similar scenarios. Candidate simulator scenarios will be identified (e.g., SBO), and the simulator environments and simulation will be configured to run an example scenario. An important element of this is developing scenarios that inform the data needs of the simulation platform. Empirical simulator studies can serve to validate predictions of the HUNTER framework, but there may also be cases where there is insufficient information available to model certain aspects of operator decision making or actions. In such cases, simulators are the primary means to model building.

# 6. CONCLUSIONS AND PATH FORWARD

## 6.1 Review of Computation-Based HRA Approach

In this report, we have outlined an approach to computation-based HRA. Computation-based HRA is similar to dynamic HRA methods except that it makes greater use of overall plant models and considers dimensions beyond time as part of its modeling. In practice, existing dynamic HRA methods like IDAC (Chang et al., 2007a) encompass many of these features. Our approach is not to reinvent suitable methods where they exist. Instead, we seek to provide a framework to combine the best HRA approaches as they apply to specific problem sets in operator performance. As such, the HUNTER approach should not be considered a new HRA method but rather a multi-method or hybrid umbrella to allow existing HRA to work in a dynamic context and beyond.

The basic approach to computation-based HRA in this report is presented in Figure 22. The two key elements of this approach are the computational engine—driven by RAVEN in the illustration—and the HUNTER HRA framework. Just as RAVEN integrates thermohydraulics and other aspects of the plant in a multi-physics model, HUNTER encapsulates various cognitive and performance modules in a multi-method structure. This multi-method is derived from cognitive models, PSFs, and various data sources, including operating experience, HRA method data tables, simulator studies, and other empirical psychological evidence. These data sources inform both the operator activities (such as information gathering, decision-making, and action) and performance outcomes. For example, the operator may have an even likelihood of either deciding to close a valve or wait for further indications in a particular plant upset condition. Available models provide the context through PSFs and the cognitive algorithms for decision outcomes. Data sources and HRA method predictions will subsequently drive the likelihood of carrying out that action. Thus, there is a twofold pass of data—decision outcomes and performance outcomes that feed into the HRA model.

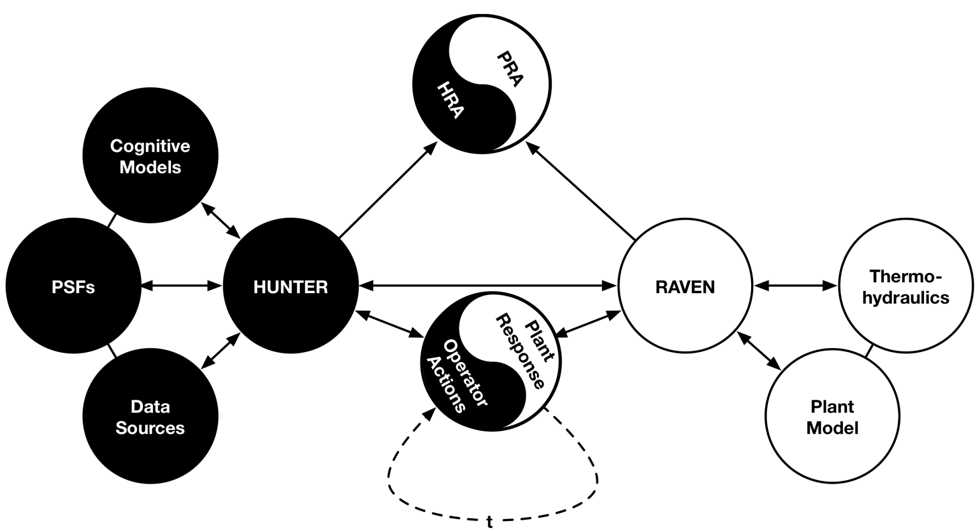


Figure 22: Framework for computation-based HRA

## 6.2 Research Plan

Overall, the research needs for our computation-based HRA approach can be translated into the following research plan:

- Develop HUNTER framework (as discussed in Chapter 4)
- Incorporate HRA elements into HUNTER (as discussed in Chapter 5)
- Integrate HUNTER with RAVEN (as discussed in Chapter 2)

This report provides only a high-level overview of the many components required for this research. Because of the multifaceted nature of this research, it is important that short-term and long-term research objectives are articulated. In the near term, this project will:

- Implement a simple case study (e.g., a flooding scenario with temporal and spatial dimensions, involving loss of key hardware systems and increased difficulties over normal operations for operators)
- Use this case study to refine the HUNTER framework and implement dynamic HRA elements
- Determine the protocol for exchanging relevant operator information with plant information through RAVEN

This simple implementation will be the basis of the next LWRS RISMC milestone report on HRA. Longer term, there is the need for:

- Integration of modules adapted from existing HRA methods into HUNTER suitable for modeling a wide cross section of plant activities
- Cataloging operator performance data and providing appropriate Bayesian updating of legacy data in HUNTER
- Validation of RAVEN-HUNTER model runs against actual crew performance data, including HRA validation studies in the HSSL
- Formal integration of HUNTER with RAVEN, including development of appropriate software libraries in RAVEN to accommodate virtual operator inputs

These activities align with immediate and planned activities in RISMC. The development of HUNTER will provide not only a robust framework for synthesizing static and dynamic HRA methods developed to date but will also provide greater fidelity on overall plant performance and risk modeling. By accounting for human activities at the plant through HUNTER, it will be possible to reduce current limitations and uncertainties in these models. HUNTER is an enabling technology to other modeling efforts currently under development. Because of the central role of the human operator in determining plant outcomes, HUNTER is an essential element of valid nuclear models.

# 7. REFERENCES


Acosta, C., & Siu, N. (1993). Dynamic event trees in accident sequence analysis: Application to steam generator tube rupture. *Reliability Engineering and System Safety*, *41*, 135-154.

Alfonsi, A., Rabiti, C., Mandelli, D., Cogliati, J., & Kinoshita, R. (2013). Raven as a tool for dynamic probabilistic risk assessment: Software overview. *Proceedings of International Conference of mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*, Sun Valley, ID.

Amendola, A., & Reina, G. (1984). *Dylam-1: A Software Package for Event Sequence and Consequence Spectrum Methodology, EUR-924, CEC-JRC*. Ispra: Commission of the European Communities

Anderson, J.R. (1993). Production systems and the ACT-R architecture. In *Rules of Mind* (pp. 1-14). Hillsdale, NJ: Lawrence Erlbaum Associates.

Baumgartner, P., & Payr, S. (Eds.), (1995). *Speaking Minds: Interviews with Twenty Eminent Cognitive Scientists.* Princeton, NJ: Princeton University Press.

Bell, B.J., & Holroyd, J. (2009). Review of Human Reliability Assessment Methods, RR679. Buxton, UK: Health and Safety Executive.

Bell, B.J., & Swain, A.D. (1983). *A Procedure for Conducting Human Reliability Analysis for Nuclear Power Plants, Final Report, NUREG/CR-2254*. Washington, DC: U.S. Nuclear Regulatory Commission.

Block, N. (1993). Troubles with functionalism. In A.I. Goldman (Ed.), *Readings in Philosophy and Cognitive Science* (pp. 231-253). Cambridge, MA: MIT Press.

Boring, R.L. (2007). Dynamic human reliability analysis: Benefits and challenges of simulating human performance. In T. Aven & J.E. Vinnem (Eds.), *Risk, Reliability and Societal Safety, Vol. 2: Thematic Topics. Proceedings of the European Safety and Reliability Conference (ESREL 2007)* (pp. 1043-1049). London: Taylor & Francis.

Boring, R.L., Agarwal, V., Joe, J.C., & Persensky, J.J. (2012). *Digital Full-Scope Mockup of a Conventional Nuclear Power Plant Control Room, Phase 1: Installation of a Utility Simulator at the Idaho National Laboratory, INL/EXT-12-26367.* Idaho Falls, ID: Idaho National Laboratory.

Boring, R., Agarwal, V., Fitzgerald, K., Hugo, J., and Hallbert, B. (2013). *Digital Full-Scope Simulation of a Conventional Nuclear Power Plant Control Room, Phase 2: Installation of a Reconfigurable Simulator to Support Nuclear Plant Sustainability, INL/EXT-13-28432*. Idaho Falls, ID: Idaho National Laboratory.

Boring, R.L., Gertman, D.I., Whaley, A.M., Joe, J.C., Tran, T.Q., Keller, M.D., Thompson, R.P.,

& Gerszewski, I.R. (2006). A taxonomy and database for capturing human reliability and human performance data. *Proceedings of the 50th Annual Meeting of the Human Factors and Ergonomics Society,* 2217-2221.

Boring, R.L., Joe, J.C., & Mandelli, D. (2015). Human performance modeling for dynamic human reliability analysis. *Digital Human Modelling 2015, Part I, Lecture Notes in Computer Science 9184*, 223-234.

Boring, R., Kelly, D., Smidts, C., Mosleh, A., and Dyre, B. (2012). Microworlds, simulators, and simulation: Framework for a benchmark of human reliability data sources. Joint Probabilistic Safety Assessment and Management and European Safety *and Reliability Conference*, 16B-Tu5-5.

Boring, R.L., Shirley, R.B., Joe, J.C., Mandelli, D., & Smith, C.L. (2014). *Simulation and Non-Simulation Based Human Reliability Analysis Approaches, INL/EXT-14-33903*. Idaho Falls, ID: Idaho National Laboratory.

Broberg, H., Hildebrandt, M., & Nowell, R.(2010) *Results from the 2010 HRA data collection at a US PWR training simulator*. HWR-981; OECD Halden Reactor Project; Halden, Norway.

Brooks, R.A. (1997). Intelligence without representation. In J. Haugeland (Ed.), *Mind Design II: Philosophy, Psychology, Artificial Intelligence* (pp. 395-420). Cambridge, MA: MIT Press.

Bye, A., Lauman, K., Braarud, P.Ø., & Massaiu, S. (2006). Methodology for improving HRA by simulator studies. *Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management (PSAM8),* 1-9.

Chalmers, D.J. (1996). *The Conscious Mind: In Search of a Fundamental Theory*. New York: Oxford University Press.

Chandler, F.T., Chang, Y.H.J., Mosleh, A., Marble, J.L., Boring, R.L., & Gertman, D.I. (2006). *Human Reliability Analysis Methods: Selection Guidance for NASA*. Washington, DC: NASA Office of Safety and Mission Assurance Technical Report.

Chang, Y.J., Bley, D., Criscione, L., Kirwan, B., Mosleh, A., Madary, T., Nowell, R., Richards , R., Roth, E.M., Sieben, S., & Zoulis, A. (2014). The SACADA database for human reliability and human performance. *Reliability Engineering and System Safety*, *125*, 117–133.

Chang, Y.J. & Lois, E. (2012). Overview of the NRC's HRA data program and recent activities. In: *Proceedings of the international conference on probabilistic safety assessment and management (PSAM 11),* Helsinki, Finland.

Chang, Y.H.J., & Mosleh, A. (2007a). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents -- Part 1: Overview of IDAC model. *Reliability Engineering & System Safety*, *92*, 997-1013.

Chang, Y.H.J., & Mosleh, A. (2007b). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents -- Part 2: IDAC performance influencing factors model. *Reliability Engineering and System Safety*, *92*, 1014–1040.

Chang, Y.H.J., & Mosleh, A. (2007c). Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents -- Part 5: Dynamic Probabilistic Simulation of IDAC Model. *Reliability Engineering & System Safety*, *92*, 1076-1101.

Churchland, P.M. (1988). *Matter and Consciousness*, *Revised Edition*. Cambridge, MA: MIT Press.

Committee on the Safety of Nuclear Installations (2012). *Simulator studies for HRA purposes*, NEA/CSNI/R(2012)1; OECD Nuclear Energy Agency; 2012, Budapest, Hungary.

Coyne, K. (2009). *A Predictive Model of Nuclear Power Plant Crew Decision-Making and Performance in a Dynamic Simulation Environment.* (Unpublished doctoral dissertation). University of Maryland, College Park.

Coyne, K. & Siu N. (2013). Simulation-based analysis for nuclear power plant risk assessment: Opportunities and challenges *Proceedings of the ANS Embedded Conference on Risk Management for Complex Socio-Technical Systems*, Washington D.C.

David, A., Berry, R., Gaston, D., Martineau, R., Peterson, J., Zhang, H., Zhao, H., Zou, L. (2012). *RELAP-7 Level 2 Milestone Report: Demonstration of a Steady State Single Phase PWR Simulation with RELAP-7, INL/EXT-12-25924*. Idaho Falls, ID: Idaho National Laboratory.

Dawson, M.R.W. (1998). *Understanding Cognitive Science.* Oxford, UK: Blackwell Publishers.

Deutsch, D. (2012, October). The very laws of physics imply that artificial intelligence must be possible. What's holding us up? *Aeon Magazine.* Retrieved from http://aeon.co/magazine/technology/david-deutsch-artificial-intelligence/

Di Pasquale, V., Miranda, S., Iannone, R., & Riemma, S. (2015). A simulator for human error probability analysis (SHERPA). *Reliability Engineering & System Safety*, *139*, 17-32.

Easterling, R.G. (1983). Appendix B: An alternative method for estimating the effects of dependence. In A.D. Swain & H.E. Guttman, *Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications, Final Report, NUREG/CR-1278*. Washington, DC: U.S. Nuclear Regulatory Commission.

Ekanem, N. J. & Mosleh, A. (2014, June). Phoenix – A model-based human reliability analysis methodology: Quantitative analysis procedure and data base. *12th Bi-Annual International Meeting of the Probabilistic Safety Assessment and Management (PSAM) Conference,* Honolulu, HI.

Electric Power Research Institute (EPRI). (1992). *SHARP1—A Revised Systematic Human Action Reliability Procedure, EPRI-101711*. Palo Alto: Electric Power Research Institute.

Flanagan, O. (1992). *Consciousness Reconsidered*. Cambridge, MA: MIT Press.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*, 3-71.

Forester, J., Dang, V.N., Bye, A., Lois, E., Massaiu, S., Broberg, H., Braarud, P.Ø., Boring, R., Männistö, Liao, H., Julius, J., Parry, G., & Nelson, P. (2012). *The International HRA Empirical Study—Final Report—Lessons Learned from Comparing HRA Methods Predictions to HAMMLAB Simulator Data, NUREG-2127*. Washington, DC: US Nuclear Regulatory Commission.

Gardiner, C.W. (2002). Handbook of stochastic methods: For physics, chemistry and the natural sciences. In H. Haken (Ed.), *Springer Series in Synergetics: Book 13*. Berlin: Springer-Verlag.

Garson, J.W. (1994). No representations without rules: The prospects for a compromise between paradigms in cognitive science. *Mind and Language*, *9*, 25-37.

Gaston, D., Hansen, G., & Newman, C. (2009). MOOSE: A parallel computational framework for coupled systems for nonlinear equations. *International Conference on Advances in Mathematics, Computational Methods, and Reactor Physics*, Saratoga Springs, NY.

Gertman, D., Blackman, H., Marble, J., Byers, J., & Smith, C. (2005). *The SPAR-H Human Reliability Analysis Method, NUREG/CR-6883*. Washington, DC: U.S. Nuclear Regulatory Commission.

Gertman, D.I., Gilmore, W.E., Galyean, W.J., Groh, M.R., Gentillon, C.D, Gillbert, B.G., & Reece, W.J. (1990). *Nuclear Computerized Library for Assessing Reactor Reliability (NUCLARR), NUREG/CR-4639, Vol. 1, Rev. 1.* Washington, DC: U.S. Nuclear Regulatory Commission.

Gibson, H., Basra, G., & Kirwan, B. (1999). Development of the CORE-DATA database. *Safety & Reliability Journal, 19*, 6-20.

Griffith, C.D. & Mahadevan, S. (2011). Inclusion of fatigue effects in human reliability analysis. *Reliability Engineering and System Safety, 96,* 1437-1447.

Groth, K., & Mosleh, A. (2009). *A Data-Informed Model of Performance Shaping Factors for Use in Human Reliability Analysis.* University of Maryland, College Park: Center for Risk and Reliability.

Groth, K. M., Smith, C. L. & Swiler, L. P. (2014). A Bayesian method for using simulator data to enhance human error probabilities assigned by existing HRA methods. *Reliability Engineering & System Safety, 128***,** 32-40.

Groth, K. M., & Swiler, L. P. (2013). Bridging the gap between HRA research and HRA practice: A Bayesian network version of SPAR-H. *Reliability Engineering & System Safety*, 115**,** 33-42.

Hakobyan, A., Aldemir, T., Denning, R., Dunagan, S., Kunsman, D., Rutt, B., & Catalyurek, U. (2008). Dynamic generation of accident progression event trees. *Nuclear Engineering and Design*, *238*, 3457–3467.

Hallbert, B., Boring, R., Gertman, D., Dudenhoeffer, D., Whaley, A., Marble, J., & Joe, J. (2006). *Human Event Repository and Analysis (HERA) System Overview, Draft Report, NUREG/CR-6903, Volume 1.* Washington, DC: U.S. Nuclear Regulatory Commission.

Helton, J.C., & Davis, F.J. (2003). Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, *81*, 23-69.

Hunt, M.M. (1993). *The Story of Psychology*. New York: Doubleday.

Institute of Electrical and Electronics Engineers (IEEE). (1997). *Guide for Incorporating Human Action Reliability Analysis for Nuclear Power Generating Stations, IEEE 1082*. New York: Institute of Electrical and Electronics Engineers.

Institute of Nuclear Power Operations (INPO). (2011). *Special Report on the Nuclear Accident at the Fukushima Daiichi Nuclear Power Station,* INPO-11-005. Atlanta: Institute of Nuclear Power Operations.

International Atomic Energy Agency (IAEA). (2003). *Means of Evaluating and Improving the Effectiveness of Training of Nuclear Power plant Personnel, IAEA-TECDOC-1358*. Vienna: International Atomic Energy Agency.

International Atomic Energy Agency (IAEA). (2004). *Use of Control Room Simulators for Training of Nuclear Power Plant Personnel, IAEA-TECDOC-1411.* Vienna: International Atomic Energy Agency.

International Nuclear Safety Advisory Group. (1992). *The Chenobyl Accident: Updating of INSAG-1*. Retrieved from http://www-pub.iaea.org/MTCD/publications/PDF/

Joe, J.C., & Boring, R.L. (2014). Individual differences in human reliability analysis. *12th Bi-Annual International Meeting of the Probabilistic Safety Assessment and Management (PSAM) Conference,* Honolulu, HI.

Joe, J.C., Shirley, R., Mandelli, D., Boring, R., & Smith, C. (in press). The development of dynamic human reliability analysis simulations for inclusion in risk informed safety margin characterization frameworks. *Proceedings of the 6th International Conference on Applied Human Factors and Ergonomics (AHFE).*

Johnson-Laird, P. (1993). *The Computer and the Mind: An Introduction to Cognitive Science*, *2nd Edition*. London: Fontana Press.

Kahneman, D. (2011). *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux.

Kloos, M., & Peschke, J. (2006). MCDET: A probabilistic dynamics method combining Monte Carlo simulation with the discrete dynamic event tree approach. *Nuclear Science and Engineering*, *153*, 137–156.

Kolaczkowski, A., Forester, J., Lois, E., and Cooper, S. (2005). *Good Practices for Implementing Human Reliability Analysis (HRA), Final Report, NUREG-1792*. Washington, DC: U.S. Nuclear Regulatory Commission.

Kruglanski, A.W., & Thompson, E.P. (1999). Persuasion by a single route: A view from the unimodal. *Psychological Inquiry*, *10*, 83-109.

Li, Y. (2013). *Modeling and Simulation of Operator Knowledge-Based Behavior*. (Unpublished doctoral dissertation). University of Maryland, College Park.

Macdonald, C. (1995). Introduction: Classicism v. connectionism. In C. Macdonald & G. Macdonald (Eds.), *Connectionism. Debates on Psychological Explanation* (pp. 3-27). Oxford: Blackwell Publishers.

Mandelli, D., & Smith, C. (2012). Adaptive sampling using support vector machines. *Proceedings of American Nuclear Society (ANS)*, *107*, 736- 738.

Mandelli, D., Smith, C., Prescott, S., Alfonsi, A., Rabiti, C., Cogliati, J., & Kinoshita, R. (2013). *Analysis of PWR SBO Caused by External Flooding Using the RISMC Toolkit, INL/EXT-14-32907*. Idaho Falls, ID: Idaho National Laboratory.

Mandelli, D., Smith, C., Riley, T., Schroeder, J., Rabiti, C., Alfonsi, A., Nielsen, J., Maljovec, D., Wang, B., & Pascucci, V. (2013). *Support and Modeling for the Boiling Water Reactor Station Black Out Case Study Using RELAP and RAVEN, INL/EXT-13-30203*. Idaho Falls, ID: Idaho National Laboratory.

Marr, D. (1982). *Vision*. San Fancisco, CA: W.H. Freeman

Meister, D. (1999). *The History of Human Factors and Ergonomics.* Mahwah, NJ: Lawrence Erlbaum.

Mosleh, A. (2014). PRA: A perspective on strengths, current limitations, and possible improvements. *Nuclear Engineering and Technology*, *46*, 1-10.

Orchanian, L.C., Smahel, T.P., Howie, D.E., & Vicente, K.J. (1996). *DURESS II User's Manual: A Thermal-Hydraulic Process Simulator for Research and Teaching.* Toronto: University of Toronto Press.

Park, J. & Jung, W. (2007). OPERA—a human performance database under simulated emergencies of nuclear power plants. *Reliab Eng Syst Saf, 92 (4*), 503–519

Pylyshyn, Z.W. (1984). *Computation and Cognition*. Cambridge, MA: MIT Press.

Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., & Tucker, P.K. (2005). Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, *41*, 1-28.

Rabiti, C., Alfonsi, A., Mandelli, D., Cogliati, J., & Kinoshita, R. (2013). Raven as a tool for dynamic probabilistic risk assessment: Software overview. *Proceedings of M & C 2013 International Topical Meeting on Mathematics and Computation, CD-ROM.* LaGrange Park, IL: American Nuclear Society.

Rabiti, C., Alfonsi, A., Mandelli, D., Cogliati, J., & Kinoshita, R. (2014). *Advanced Probabilistic Risk Analysis Using RAVEN and RELAP-7*, INL/EXT-14-32491. Idaho Falls, ID: Idaho National Laboratory.

Rasmussen, J. (1986). Development and testing of a model for simulation of process operator response during emergencies in nuclear power plants. In *Proceedings of the International Topical Meeting on Advances in Human Factors in Nuclear Power Systems*, Knoxville, TN.

RELAP5 Code Development Team (2012). *RELAP5-3D Code Manual, INL/EXT-12-00834*. Idaho Falls, ID: Idaho National Laboratory.

Robertson Museum and Science Center. (2000). *The Link Flight Trainer: A Historic Mechanical Engineering Landmark.* New York: American Society of Mechanical Engineers.

Rumelhart, D. E., McClelland, J. L., & the PDP research group. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume I.* Cambridge, MA: MIT Press.

Russell, S.J., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Ryan, D., Winningham, K., Metzroth, K., Aldemir, T. & Denning, R. (2009). Passive heat removal system recovery following an aircraft crash using dynamic event tree analysis. *Proceedings of American Nuclear Society (ANS)*, *100*, 461- 462.

Smith, C., Rabiti, C., & Martineau, R. (2011). *Risk Informed Safety Margin Characterization (RISMC) Pathway Technical Program Plan, INL/EXT-11-22977*. Idaho Falls, ID: Idaho National Laboratory.

Skjerve, A.B., & Bye, A. (2010). *Simulator-based human factors studies across 25 years: the history of the Halden man-machine laboratory*, Springer, London (2010).

U.S. Nuclear Regulatory Commission (1987). *BWR and PWR Off-Normal Event Descriptions*, *NUREG-1291*.Washington, DC: U.S. Nuclear Regulatory Commission.

U.S. Nuclear Regulatory Commission (1990). *Severe accident risks: an assessment for five U.S. nuclear power plants*, *NUREG-1150*. Division of Systems Research, Washington, DC: U.S. Nuclear Regulatory Commission.

U.S. Nuclear Regulatory Commission (2000). *Technical Basis and Implementation Guidelines for A Technique for Human Event Analysis (ATHEANA), NUREG-1624, Rev. 1*.Washington, DC: U.S. Nuclear Regulatory Commission.

U.S. Nuclear Regulatory Commission (2014). *The International HRA Empirical Study: Lessons Learned from Comparing HRA Methods Predictions to HAMMLAB Simulator Data, NUREG-2127*, "Washington, DC: U.S. Nuclear Regulatory Commission.

U.S. Nuclear Regulatory Commission (2014). "Backgrounder on the Three Mile Island Accident", http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html.

Whaley, A.M., Xing, J., Boring, R.L., Hendrickson, S.M.L., Joe, J.C., Le Blanc, K., and Lois, E. (2012). *Building a Psychological Foundation for Human Reliability Analysis, NUREG-2114.* Washington, DC: US Nuclear Regulatory Commission.