HUMAN UNIMODEL FOR NUCLEAR TECHNOLOGY TO ENHANCE RELIABILITY (HUNTER): A FRAMEWORK FOR COMPUTATIONAL-BASED HUMAN RELIABILITY ANALYSIS

Ronald Boring,[1] Diego Mandelli,[1] Martin Rasmussen,[2] Sarah Herberger,[1] Thomas Ulrich,[1]
Katrina Groth,[3] and Curtis Smith[1]

[1] *Idaho National Laboratory: PO Box 1625, Idaho Falls, Idaho 83415, USA*
[2] *NTNU Social Research, Dragvoll Allé 38 B, 7491 Trondheim, Norway*
[3] *Sandia National Laboratories, 1515 Eubank, Albuquerque, New Mexico 87123, USA*

A computation-based human reliability analysis framework called the Human Unimodel for Nuclear Technology to Enhance Reliability (HUNTER) has been developed as part of the Risk Informed Safety Margin Characterization (RISMC) pathway within the U.S. Department of Energy's Light Water Reactor Sustainability Program that aims to extend the life of the currently operating fleet of U.S. commercial nuclear power plants. HUNTER is a flexible hybrid approach that functions as a framework for dynamic modeling, including a simplified model of human cognition—a virtual operator—that produces relevant outputs such as the human error probability (HEP), time spent on task, or task decisions based on relevant plant evolutions. HUNTER is the human reliability analysis counterpart to the Risk Analysis in a Virtual ENvironment (RAVEN) framework used for dynamic probabilistic risk assessment. Although both RAVEN and HUNTER are under various stages of development, this paper presents a successfully integrated and implemented RAVEN-HUNTER initial demonstration. The demonstration centers on a station blackout scenario, using complexity as the sole virtual operator performance-shaping factor (PSF). The implementation of RAVEN-HUNTER can be readily scaled to other nuclear power plant scenarios of interest and will include additional PSFs in the future.

I. INTRODUCTION

This paper presents an application of a computation-based human reliability analysis (CBHRA) framework called the Human Unimodel for Nuclear Technology to Enhance Reliability.[1] A *unimodel*—the *U* in HUNTER—is a simplified cognitive model. Thus, HUNTER represents a simplified cognitive model or a collection of simplified cognitive models to support dynamic risk analysis. HUNTER is a hybrid approach built on past work from cognitive psychology, human performance modeling, and human reliability analysis (HRA). Using these research fields as background, HUNTER functions as a simplified model of human cognition—a virtual operator—that, when combined with a computation engine such as a thermal-hydraulics based nuclear power plant simulation model, can produce outputs such as the human error probability (HEP), time spent on task, or task decisions based on relevant plant evolutions.

HUNTER is flexible in terms of which inputs and cognitive evaluations are used and what it produces. HUNTER has been developed not as a standalone HRA method but rather as a framework that ties many HRA methods together. HUNTER then in turn applies a dynamic risk assessment of human activities and serves as an interface between HRA and other aspects of the dynamic modeling, such as thermal-hydraulic code, as part of overall probabilistic risk assessment (PRA).

HUNTER is the HRA counterpart to the Risk Analysis in a Virtual ENvironment (RAVEN) framework in PRA,[2] as depicted in Fig 1. Although both RAVEN and HUNTER are under various stages of development, a successfully integrated and implemented RAVEN-HUNTER demonstration is presented in this paper. The demonstration centers on a station blackout scenario, but the implementation of RAVEN-HUNTER is scalable to other nuclear power plant scenarios.

HUNTER was created with the goal of including HRA in areas where it has not been represented thus far and to reduce uncertainty by accounting for human performance more accurately than many current HRA approaches. While we have adopted particular methods to build an initial model, the HUNTER framework is intrinsically flexible to new modules that achieve particular modeling goals. Computation-based HRA in HUNTER does not consist of a single HRA model or method;

rather, it can encompass a number of different HRA approaches that account for different aspects of human performance. A goal of HUNTER is, in fact, to "dynamicize" legacy HRA approaches wherever feasible.

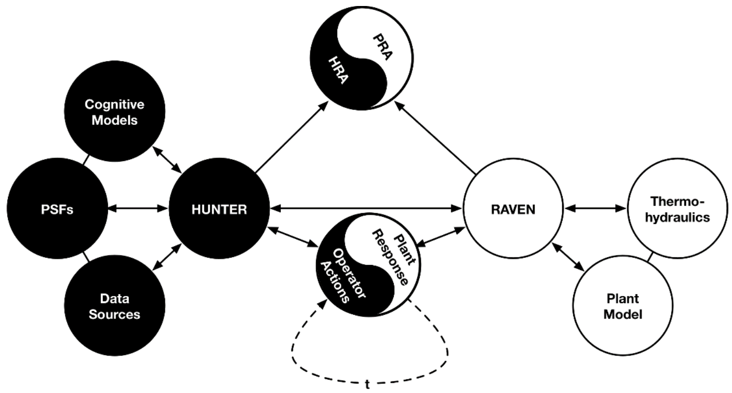

Fig 1. Framework for computation-based HRA (from Ref. 1)

The HUNTER project is part of the Risk Informed Safety Margin Characterization (RISMC) research pathway within the U.S. Department of Energy's Light Water Reactor Sustainability (LWRS) program, which aims to extend the life of the currently operating fleet of U.S. commercial nuclear power plants. HUNTER has the potential to model risk more accurately across a greater range of scenarios than has been possible with conventional HRA approaches. Additionally, HUNTER provides a crucial connection between RAVEN and human performance, which extends the utility of that modeling code. As such, HUNTER ultimately aims to ensure the continued safety and reliability of currently operating nuclear power plants.

II. COMPUTATION-BASED HUMAN RELIABILITY ANALYSIS

In a traditional (or static) HRA, the human reliability analyst determines the quantification by choosing the most suited task type and/or appropriate PSFs, which is then used in an equation to estimate the HE. This somewhat simplified description of HRA may falsely provide the impression that performing an HRA is a quick and easy task in which the analyst simply makes a few choices to produce an HEP value. A properly executed traditional HRA relies on a solid qualitative data collection and qualitative data analysis.

Mosleh[3] and Coyne and Siu[4] 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. Here we explore the human side of dynamic PRA. The key elements of dynamic or computation-based HRA 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 the present research is to achieve a high fidelity causal representation of the role of the human operator at the plant. By better 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 currently not clearly understood or predicted, e.g., unexampled events such as severe accidents. The ability to simulate the role of the human operator complements and, indeed, greatly enhances other PRA modeling efforts.

The approach of CBHRA relies on the creation of a virtual operator that is interfaced with a realistic plant model that can accurately simulate plant thermal-hydraulic physics behavior.[1] Ultimately, the virtual reactor operator should consist of comprehensive cognitive models comprised of artificial intelligence, though at this time a much more simplified operator model is used to simulate performance of a typical operator. CBHRA is a merger between an area where HRA has previously been represented—probabilistic risk models—and an area where it has not—realistically simulated plant models through mechanistic thermal-hydraulic multi-physics codes. Through this approach, it is possible to evaluate a much broader spectrum of scenarios, both those based on previous experience and those that are unexampled, i.e., that have not been assessed with static HRA.

This is a promising path to advance the methodology of HRA, but there are numerous challenges that must be overcome before a fully functioning plant simulation including a virtual operator model is realized. In CBHRA, a scenario can be rapidly simulated thousands of times, which renders individual subjective evaluations by a human reliability analyst during each simulation run impractical. Unfortunately, most of the PSFs in current HRA methods are operationalized and described in a way that suits subjective evaluations from the analyst, which presents challenges to translate the static optimized methods to a coding scheme that can automatically and dynamically set the PSF at the correct level during simulation runs.

While it is tempting simply to script human actions at the nuclear power plant according to operating procedures, there remains considerable variability in operator performance despite the most formalized and invariant procedures to guide activities. 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, secondly, 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—the PSFs—will only result in uncertain outcomes.

Boring,[5] among others, explains the conceptual shift from static HRA to computation-based 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. CBHRA 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. The dynamic approach in PRA has proved especially useful for modeling beyond design basis accidents, where not all failure combinations and 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 static 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.

III. MOOSE, RAVEN, AND RELAP-7

RAVEN acts as the computational engine behind HUNTER. A real reactor system is very complex and may contain thousands of different physical components. Therefore, it is impractical to preserve real geometry for the whole system. Instead, simplified thermal-hydraulic models are used to represent the major physical components and describe major physical processes. The manipulation of variables is performed by two components of the RAVEN simulation controller:

- RAVEN control logic is the system control logic of the simulation where, based on the status of the system, it updates the status/value of the controlled parameters
- RAVEN/RELAP-7 interface updates and retrieves component variables according to the control logic
- Auxiliary variables are user to defined simulation specifications that may be needed to limit the simulation.

From a mathematical point of view, auxiliary variables are the ones that guarantee the system to be Markovian. The set of auxiliary variables also includes those that monitor the status of specific control logic set of components and simplify the construction of the overall control logic scheme of RAVEN.

RELAP-7 thermal-hydraulics code is designed to be the main reactor system simulation toolkit for the RISMC Pathway of the LWRS Program. RELAP-7 code development takes advantage of the progress made in the past several decades to achieve simultaneous advancement of physical models, numerical methods, and software design. RELAP-7 uses the Multi-Physics Object-Oriented Simulation Environment (MOOSE) framework for solving computational engineering problems in a well-planned, managed, and coordinated way (see Fig. 2). This allows RELAP-7 development to focus strictly on system analysis-type physical modeling and gives priority to retention and extension of RELAP5's multidimensional system capabilities.

RAVEN is a software framework that acts as the control logic driver for the thermal-hydraulic code RELAP- 7. RAVEN is also a multi-purpose PRA code that allows for probabilistic analysis of complex systems. It is designed to derive and actuate the control logic required to simulate both plant control system and operator actions and to perform both Monte-Carlo sampling of random distributed events and dynamic branching-type analyses. The RAVEN statistical framework is a recent add-on to the overall RAVEN package that allows the user to perform generic statistical analysis.

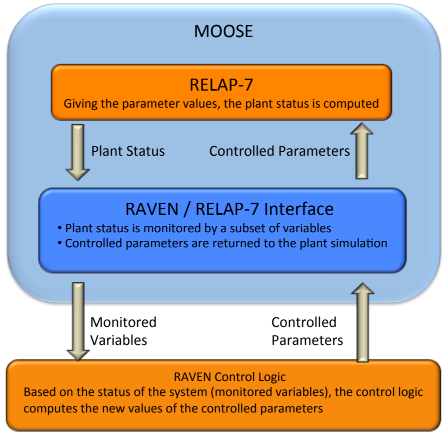

Fig. 2. The MOOSE, RAVEN, & RELAP-7 simulation approach

IV. HUMAN RELIABILITY SUBTASK PRIMITIVES: GOMS-HRA

One of the challenges in CBHRA is the fact that most HRA methods quantify at the overall task level, while subtask quantification will often be required for the CBHRA to best follow the scenario as it develops. In an attempt to overcome this challenge, we developed a new HRA approach through categorizing subtasks and linking them to human error probabilities.[6] The purpose of developing this new approach was to allow us to anchor our analyses on subtasks as required by CBHRA, because existing HRA methods did not—in the authors' views—adequately address subtask analysis.

The Goals, Operators, Methods, and Selection rules (GOMS) method was first developed by Card, Moran, and Newell.[7] Goals represent the high level tasks the human seeks to complete, Operators are the available actions the human can take, Methods are the steps or subgoals the human takes toward completing Goals, and Selection rules are the decisions the humans make. GOMS has been used extensively in human factors as a way to model proceduralized activities. It shares underpinnings with task analysis in that it breaks human actions into a series of subtasks. By cataloging particular types of actions, it is possible to predict human actions or task durations. GOMS has also been used in the human factors community to model user interactions with human-computer interfaces. The predictive abilities of GOMS provide an alternative to user studies, but GOMS has been criticized for being time consuming and labor intensive to model.

GOMS-HRA features a selection of task level primitives, representing the most basic action types by operators:

- *Actions* (A)—Performing required physical actions on the control boards (A_C) or in the field (A_F)
- *Checking* (C)—Looking for required information on the control boards (C_C) or in the field (C_F)
- *Retrieval* (R)—Obtaining required information on the control boards (R_C) or in the field (R_F)
- *Instruction Communication* (I)—Producing verbal or written instructions (I_P) or receiving verbal or written instructions (I_R)
- *Selection* (S)—Selecting or setting a value on the control boards (S_C) or in the field (S_F)
- *Decisions* (D)—Making a decision based on procedures (D_P) or without available procedures (D_W)

Procedure steps may be decomposed into GOMS task level primitives. The procedure level primitive used within each procedure step represents a cluster of actions that must occur in the proper sequence in order for the operator to successfully complete the step. These procedure level primitives can be decomposed into sequences of task primitives. The sequence of task level primitives repeats iteratively until the desired value or state is achieved and the step is concluded. The task level primitives from GOMS-HRA were mapped for each procedure step in order to support the estimation of both completion times and HEP values for each step (see Table I).

Table I. Generic procedure level primitive mapping to task level primitives

Procedure Level Primitive	Definition	Task Level Primitive	Mapping Notes
Determine	Calculate, find out, decide, or evaluate.	C_C or R_C	Information type dependent
Ensure	Perform a comparison with stated requirements and take action as necessary to satisfy the requirements.	C_C or R_C and/or A_C and/or S_C	Information and control action type dependent
Initiate	Begin activity function or process.	A_C	-
Isolate	Separate, set apart, seal off, or close boundary.	A_C	-
Minimize	Make as small as possible.	S_C	-
Open	Change the physical position of a mechanical device to allow flow through a valve or prevents passage of electrical current.	A_C	-
Verify	Observe an expected condition exist - no actions to correct	C_C, R_C	Information type dependent

Table I depicts the procedure level primitives identified in the simulation log data that were used to decompose the procedure level primitives into task level primitives. The procedure level primitives are generically defined in this table since the object on which the procedure level primitive operates is not defined.

V. COMPLEXITY: MODELING PERFORMANCE SHAPING FACTORS

As mentioned, quantification is fundamentally different between traditional static HRA and CBHRA. The largest difference will be that the decisions made by a human reliability analyst in traditional static HRA will be modeled by a virtual operator in CBHRA. The decisions of the virtual operator will, however, be influenced by many of the same aspects as shape the traditional analysis. Before a scenario is simulated, potential tasks will have to be modeled, and this modeling will contain categorization elements that are similar to the task type and PSF choices that are made in traditional static HRA.

Complexity is included in most HRA methods as part of the quantification of the HEP. This fits well with our intuitive understanding of complexity and the role it can have in the likelihood of successfully conducting a task. Complexity is, however, a multifaceted concept and there are challenges in finding or creating a fitting operationalization. In Rasmussen, Standal, and Laumann,[8] a task complexity model containing six factors (goal-, size-, step-, dynamic-, structure- and connection complexity) was presented. The work in Ref. 8 initially examined 13 complexity factors, with seven subsequently being excluded (procedure-, temporal-, knowledge-, human-machine interface (HMI)-, interaction- and variation complexity and uncertainty). The main reason for the exclusion was overlap with other PSFs. The thirteen-factor model was not clearly orthogonal.

As CBHRA allows a scenario to develop instead of following a scripted path, complexity is not included in the HRA model in the same way it is in static HRA. Instead, some of the simulations will develop in a way that complexity expands, while others will follow paths with reduced complexity. This will allow CBHRA to better model scenarios that could develop in many ways, with numerous different correct response option pathways and various acceptable outcomes, or include richer pathways including aspects such as recovery actions, in which steps must be redone correctly to achieve the desired outcome.

A primary advantage of dynamically modeling complexity in CBHRA is that a task can have more than one output. Instead of only providing a direct contribution to the HEP for a single event tree, it can provide an influence to path choices and reshape the event tree, which will also elicit influence on the dynamically changing HEP, as some paths will lead to additional non-desired results. Complexity will also elicit influence on the time spent on the task, which will also influence the HEP dynamically as most scenarios possess some finite time limit for actions to be effective. CBHRA allows for the inclusion of different degrees of variance in time spent on a task. This is relevant to the inclusion of complexity as it is likely that complex tasks have more variance in time spent than a non-complex task as depicted in Fig. 3.

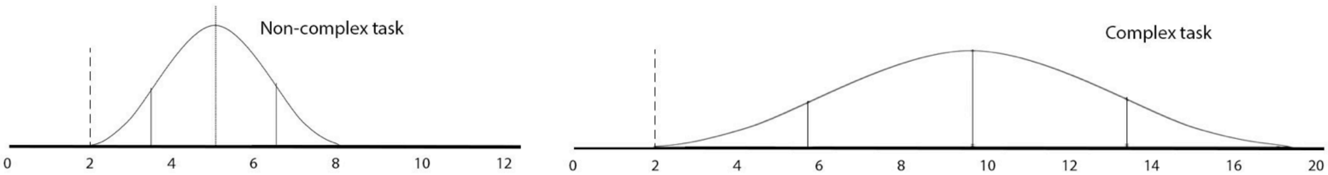

Fig. 3. Hypothetical time spent on a non-complex and complex task with minimum required time of two minutes.

Another important advantage of modeling complexity in CBHRA is increased capability to appropriately calibrate the method by using empirical data. If operational or simulator data are available for a scenario, it could be used to evaluate the values in a CBHRA method. Real life data on both near misses and major accidents are fortunately scarce, but both simulator data and databases that include human actions could be used in calibrating the virtual operator.

VI. QUANTIFYING THE HUMAN ERROR PROBABILITY

Quantification of the human error probability is one of the primary objects of HRA as it is used to assess the performance of human actions. Quantifying errors typically includes providing a probabilistic description of the likelihood for the errors to occur. The quantification process makes use of nominal HEPs, which are base error likelihoods for a generic task type, such as closing a valve. These nominal HEPs are intentionally formulated to describe generic human actions to support their application to many different tasks. Generic HEPs serve as the basic toolset of HRA quantification in which the context of the task can be layered upon to tailor these generic HEPs to highly specific tasks. Since errors occur within the context of the system and operating situation, PSFs capture the nuances of the specific task and modify the nominal HEP by integrating these contextual factors that affect performance. The multiplication of the generic nominal HEPs and the task specific PSFs yields the overall HEP value. PSFs can both improve or hinder operator performance. The task specific overall HEP value provides a comprehensive quantification of the task and can then be used to make risk related decisions.

One primary goal of the HUNTER approach is to support the ability to autocalculate HEPs based on contextual information. Autocalculation of the overall HEPs is needed to capture the dynamics of human error while the simulation is running. For example, we are currently modeling the effects of complexity as it evolves dynamically. As complexity increases, so should the HEP. Importantly, complexity changes as the modeled event progresses and evolves by increasing or decreasing the HEP for any subtask or slice of time accordingly. The change occurs relative to the nominal HEP value. Indeed, one of the primary reasons for decomposing subtasks into a GOMS structure is to define the Operators as the basis for the HEP. These Operators correspond to nominal HEP values, which can be modified by PSFs like complexity.

A reasonable starting point for quantifying the GOMS-HRA Operators is the original HRA method, THERP.[9] THERP uses template matching in which the analyst matches the current subtask being analyzed to similar subtasks found in the method worksheets. THERP, unlike most other HRA methods, is subtask based, and it aligns to the level of analysis required for quantifying the GOMS-HRA Operators.

Written or implied procedural steps form the subtasks modeled in dynamic HRA. Although the degree of strict procedural adherence by nuclear power plant crews may be a matter for some debate, the procedures serve as mileposts for crew actions. Furthermore, for modeling purposes, the procedure steps serve to document the solution path, which is advantageous to represent crew actions within the modeling simulation. Thus, in order to model crew behavior dynamically, procedure steps are coded into the dynamic model. The value of GOMS-HRA is that by coding each step as an Operator, it is possible to imbue the model with additional information that makes HRA possible. Each Operator classifies the type of action being performed, In short, Operator coding with GOMS-HRA becomes the skeleton to which other model elements are affixed.

VII. SIMULATION CASE STUDY: STATION BLACKOUT

Typically, commercial nuclear power plants make use of external alternating current (AC) electrical power sources. Even if the reactor is not critical, the residual heat removal systems require AC power to disperse heat generated by the nuclear core. Loss of offsite power (LOOP) events refer to the situations in which the external AC electrical power source for the plant are rendered unavailable. LOOP events are categorized based on their initiating event. Plant centered LOOP events occur anywhere within the plant up to the auxiliary or station transformers.

The specific station blackout event modeled in this simulation represents a prototypical station blackout event. After the initial LOOP event, a reactor trip triggers, which prompts the operators to enter into an emergency operating procedure. During the post trip actions procedure, the operators perform a number of plant diagnostic steps to ensure the plant is operating within safety envelopes. First they confirm the reactor successfully tripped by verifying a downward trend in reactor power. The operators then confirm the turbine has tripped and the main output breakers have opened. At this point operators' efforts turn toward confirming the safety systems are functioning properly, which includes assessing that the reactor coolant system inventory is sufficient, ensuring at least one recirculating coolant pump is in operation, and residual heat removal is capable of dissipating heat from the recirculated coolant. Lastly, the operators check the integrity of containment by verifying no radiation alarms are present and assessing containment pressure and temperatures. An outline for the timeline of the scenario follows in Fig. 4.

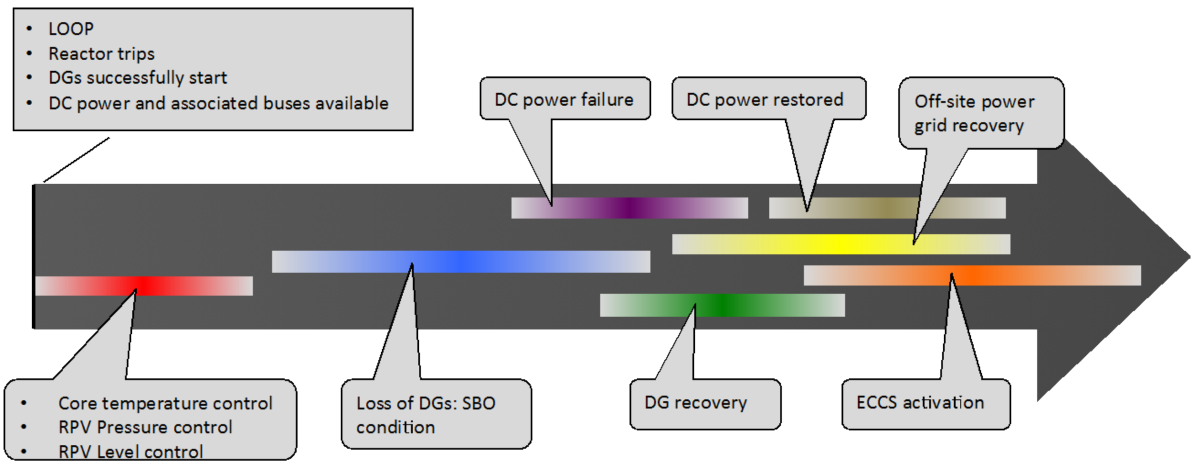

Fig. 4. Sequence of events for the SBO scenario considered

For each of these parameters we found the appropriate probability distribution function in order to evaluate core damage probability. Core damage is reached when the maximum clad temperature in the core reaches its failure temperature (2200° F). To analyze the risk associated with a station blackout, the GOMS-HRA method was applied. The GOMS-HRA method entails decomposing procedure steps into task primitives, which are then used to calculate completion time and HEP values for each procedure step. The completion time and HEP values were then input to the RAVEN model to simulate human error events and their outcomes in relation to plant thermal-hydraulics. In order to analyze a scenario, such as the station blackout event, and calculate the nominal HEP and task timing values, the procedure must be evaluated at the procedure level and then at the task level. The procedures included in this simulation are based on the post trip action and station blackout procedures.

Table II. SBO Step 5 showing mapping of Ensure procedure level primitive.

SBO	5	-	Minimize rector coolant system leakage	Minimize[1]	-
SBO	5	a	Ensure letdown is isolated	Ensure	Cc
SBO					Ac
SBO	5	b	Ensure reactor coolant pump controlled bleedoff is isolated	Ensure	Cc
SBO	5	c	Ensure reactor coolant system sampling is isolated	Ensure	Cc

Proprietary procedures cannot be publicly disseminated; however, Table II contains an example procedure step and serves to provide an overview of how a step is mapped to the procedure level and task level primitive. To reiterate the process, two mappings are involved:

- The plant procedures are classified in terms of procedure level primitives.
- These procedure level primitives are comprised of task level primitives from GOMS- HRA.

Because there is a high degree of nuclear industry consensus on terminology in operating procedures, the procedure level primitives represent commonly and consistently deployed types of activities. It is therefore possible to create a universal mapping of GOMS-HRA task level primitives to the procedure level primitives. This universal mapping affords the opportunity for reuse of the building blocks in HUNTER across different analyses.

The procedures are an approximation of the actual series of events that would unfold during the scenario, although this reduces some of the realism captured in the simulation. Furthermore, this is the first attempt at performing an integrative HRA model with dynamic HEPs and corresponding thermal-hydraulic computations, which was made possible by restricting the scope of the simulation. To illustrate this analysis further, station blackout procedure 5a stating "Ensure let down is isolated" is in Table II. The procedure level primitive in this step is defined as the verb, *Ensure*. Ensure could be decomposed into different task level primitives, so the context of the procedure step, in this case letdown isolation, must be evaluated to determine which of the task level primitives are applicable. In this instance, the valve positions are a status indicator with a simple state control as opposed to a continuous numerical value setting. As a result, this procedure level primitive translates to the task level primitives of C_C and A_C, definitions in Table I.

The procedure steps for the SBO procedures were mapped to procedure and task level primitives. Following the analysis of the procedures to map procedure level and task level primitives, timing data were estimated for each procedure step as derived from GOMS-HRA. Additionally, the procedure steps were aligned with the two primary events in which the LOOP occurs and the loss of diesel generators (LODG) and loss of battery (LOB) during the station blackout event.

VIII. CONCLUSIONS

HRA is but one part of the larger PRA framework. HRA interacts with the PRA model; however, HRA has often been performed as a standalone analysis. HUNTER provides the possibility to reduce this disconnect by interfacing HRA and PRA into a single RAVEN-HUNTER framework capable of dynamic simulation based modeling. This approach should not be seen as simply replacing traditional HRA with a new modeling form of HRA, but rather as a tool to better integrate human performance into areas of risk analysis where it has not been included thus far. As the demonstration in this report is a simplified test case, the full capabilities of HUNTER are not realized. HUNTER can model many more features when additional PSFs are incorporated, detailed aspects of the plant parameters are included, and the scenarios become more diverse and contain several paths and possible end states.

This demonstration has also shown how the GOMS-HRA approach can be used to decompose a scenario into standardized units of task level primitives. This allows for quantification at a level where autopopulating PSFs is possible and provides consistency in how a scenario is decomposed and quantified, which is something that has been previously lacking in HRA; however, this aspect is a critical part of a computationally based approach to HRA. This dynamic approach can be used to provide a more comprehensive image of risk changes throughout the unfolding of an event as opposed to the snapshot of a static event captured with traditional HRA.

This is the initial proof of concept; thus, a number of concessions were necessary to ensure this project achieved reasonable results without unduly spreading our efforts across overly ambitions research aims. A fully comprehensive simulation of the operator and the entire gamut of performance behaviors was beyond the scope of this research, but future efforts are underway to refine the methods. As a result, a number of limitations exist.

First, the level of detail in terms of actions within the procedures was restricted to systems of functionally related components as opposed to specific components. Another primary limitation concerns the PSFs used for quantification of human error in the model. This work only considered complexity as inputs to calculate the overall HEP within each time step. The HUNTER modelling approach should be capable of additional PSFs with little modification.

In this initial demonstration of HUNTER, the model of the operator consisted of a single PSF and spanned only a single scenario. Future research in HUNTER aims to move toward improving the HUNTER framework to the level in which a plant PRA model can be dynamically simulated. Dynamically modeling a plant PRA entails a large scale effort comprised of simulating accident sequence progressions, plant systems and components, and operator actions. To support this functionality, future work on HUNTER will incorporate more scenarios and the necessary procedures to support the operator models. Additionally, the operator cognitive model will be enhanced by incorporating additional PSFs to capture a more accurate portrayal of the operator and human error likelihoods during scenario evolutions.

ACKNOWLEDGMENTS AND DISCLAIMER

We express our sincere thanks for textual reviews and inputs from Gordon Bower, Nancy Lybeck, Kateryna Savchenko, and Jeff Einerson at Idaho National Laboratory. This work of authorship was prepared as an account of work sponsored by Idaho National Laboratory, an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights. Idaho National Laboratory is a multi-program laboratory operated by Battelle Energy Alliance LLC, for the United States Department of Energy under Contract DE-AC07-05ID14517.

REFERENCES

1. Boring, R.L., Mandelli, D., Joe, J., Smith, C., & Groth, K. (2015). A Research Roadmap for Computation-Based Human Reliability Analysis, INL/EXT-15-36051. Idaho Falls: Idaho National Laboratory.
2. Rabiti, C., Alfonsi A., Mandelli, D., Cogliati, J., Martinueau, R., & Smith, C. (2013). Deployment and Overview of RAVEN Capabilities for a Probabilistic Risk Assessment. Demo for a PWR Station Blackout, INL/EXT-14-33903. Idaho Falls: Idaho National Laboratory.
3. Mosleh, A. (2014). PRA: A perspective on strengths, current limitations, and possible improvements. Nuclear Engineering and Technology, 46, 1-10.
4. 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.
5. Boring, R. L. (2007). Dynamic human reliability analysis: Benefits and challenges of simulating human performance. Risk, Reliability and Societal Safety, 2, 1043-1049.
6. Boring, R.L., & Rasmussen, M. (2016, in print). GOMS-HRA: A method for treating subtasks in dynamic human reliability analysis. *Proceedings of the 2016 European Safety and Reliability Conference.*
7. Card, S., Moran, T., & Newell, A. (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.
8. Rasmussen, M., Standal, M.I. & Laumann, K., (2015). Task complexity as a performance shaping factor: A review and recommendations in Standardized Plant Analysis Risk- Human Reliability Analysis (SPAR-H) adaption. Safety Science, 76, 228–238.
9. Swain, A.D., Guttman, H.E. (1983). Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications, NUREG/CR-1278. Washington, DC: U.S. Nuclear Regulatory Commission.