Human Error, Reliability, Resilience, and Performance, Vol. 82, 2023, 1–12
https://doi.org/10.54941/ahfe1003548

Human Error and Performance Modeling With Virtual Operator Model (HUNTER) Synchronously Coupled to Rancor Microworld Nuclear Power Plant Simulator

Roger Lew[1], Ronald L. Boring[2], and Thomas A. Ulrich[2]

[1]University of Idaho, Moscow, ID 83843, USA
[2]Idaho National Laboratory, ID 83415, USA

ABSTRACT

HUNTER is the Human Unimodel for Nuclear Technology to Enhance Reliability. HUNTER is a virtual operator for a nuclear power plant (NPP). The virtual operator follows procedures and estimates the reliability and timing of real operators. The model utilizes dynamic human performance shaping factors to model performance (Boring et al., 2016). Unlike traditional human reliability analysis (HRA) modeling HUNTER uses a dynamic version of SPAR-H to calculate performance shaping factors (PSFs) based on evolving plant conditions. HUNTER accomplishes this by using task level Goals-Operators-Methods-Selection rules (GOMS)-HRA as the operator walks through procedure steps. In this paper we show how HUNTER was tightly coupled to a NPP Microworld called Rancor.

Keywords: Human reliability analysis, Human performance modeling, Risked informed systems analysis, Nuclear power, Safety

INTRODUCTION

A new suite of probabilistic risk assessment (PRA) tools called HUNTER-Rancor ties a dynamic human reliability model to the a simplified nuclear reactor simulator called Rancor. The resulting model has a virtual operator capable of monitoring and executing actions in virtual power plant. HUNTER-Rancor is an an addition to the HUNTER modeling framework (Boring, Lew, Ulrich, Park, 2023). Rancor is a simplified NPP simulator that was developed for human factors nuclear power research but has found utility for other applications. This version of the model synchronously couples with HUNTER. The synchronous coupling enables the virtual operator model and rancor plant simulator. The virtual operator can check plant conditions and execute actions. The primary benefit of synchronously coupling the two models is that the integrated model provides higher simulation fidelity and more accurate temporal dynamics. The synchronously coupled model was developed for RISMC for DOE's Light Water Reactor Sustainability project as PRA (probabilistic risk assessment) toolkit.

THE RANCOR SIMULATOR MICROWORLD

Idaho National Laboratory and the University of Idaho are co-developers of the Rancor Nuclear Power Plant Simulator, i.e., Rancor. As a "microworld simulator" the processes of a nuclear power plant are represented in a simplified manner (Ulrich, 2017).

Rancor runs on Microsoft Windows and built with Microsoft's WPF with.NET Framework 4.8. The raison d'etre of Rancor is to have a microworld process control simulator that could be used by naïve participants (e.g., students). To support engineering research and other applications the process model from the Rancor simulator was implemented in the Python. Since the HUNTER model is also implemented in Python integrating Rancor with HUNTER the two were a natural candidate for synchronous coupling. In the implementation the Rancor model is ran as an instance from within the HUNTER model. The HUNTER model takes the administrative duties of initializing, stepping, and cleaning up the model. As such the HUNTER mode is a simulator in the loop model. If we did not utilize a simulator model written in/for Python we would have the additional hassle of needing an application programming interface (API) or remote procedure call (RPC) to interface the two models.

The WPF version of Rancor has a modern Human Machine Interface, while the Python version of Rancor has simple but effective Textual User Interface (TUI) built with a new UI library from Textualize.io. Even though the TUI is very minimal and console based it allows the model to be ran interactively.

Python is generally not known for its speed, but the simplicity of reduced order model implemented in Rancor provide the ability to run much faster than normal speed. Full-scope nuclear power plant simulators have high fidelity but actually cannot run significantly faster than normal speed. The Rancor model runs at over 100x normal speed on a commodity PC.

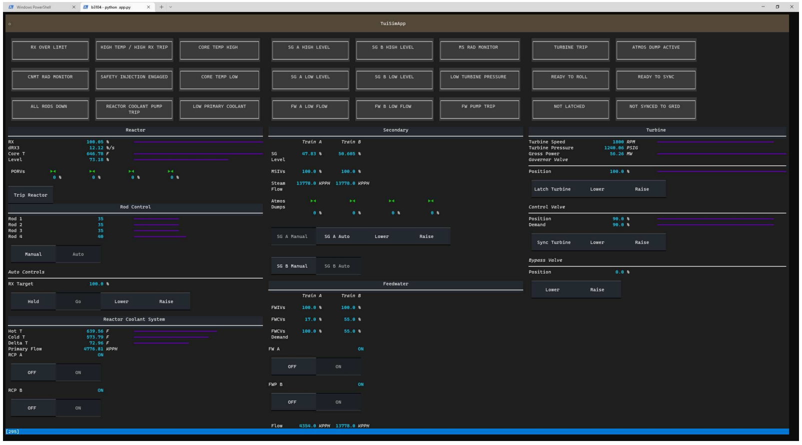

Figure 1: Textual user interface for Rancor Python model running in display terminal mode.

HUNTER-RANCOR IMPLEMENTATION

As previously noted, Python was selected to implemented the synchronous model. Notable the authors are found of the expressiveness of Python and the batteries included with Python's every growing scientific stack (Numpy, SciPy, Matplotlib, Pandas). Rancor has been used in multiple experiments over several years and has a growing corpus of plant scenarios (e.g. startup, loss of feedwater, steam generator tube rupture), procedures, and human operations data. The application utilizes predefined procedures as inputs, encompassing all the essential data elements required for the execution of proceduralized tasks, which are based on the simulated state of the nuclear plant. The state of the plant is employed to assess the logical aspects of each procedural step. Additionally, a dynamic Human Reliability Analysis (HRA) module evaluates the completion of each step by the virtual operator. This HRA module takes into account the simulation context to calculate durations of completion, Human Error Probabilities (HEPs), and determine the success or failure of each step.:

1. ***Step***—refers to a concise group of activities that serves as the fundamental organizational unit. The procedures are structured in a manner that aims to ensure closure, meaning that the success or failure of a step leads to branching either to another step within the same procedure or transitioning to a different procedure. This design allows for seamless continuity and interconnectedness between steps in the procedures.
2. ***Preconditions***—are conditions that need to be met in order to progress to the next element within a step. These conditions can be composed of multiple individual conditions, employing various types of logic such as "any" or "all" conditions needing affirmation. Precondition fulfillment can initiate a specified action, or the preconditions themselves can encompass the entire step, serving as diagnostic steps within a procedure.
3. ***Actions***—pertains to the actions undertaken by an operator to manipulate the state of a component or system. These actions may have preconditions that need to be satisfied before they can be executed.
4. ***Postconditions***—are conditions that need to be verified after an action has been executed within a step. These postconditions reflect the response of the plant to the operator's action. It is necessary for the postconditions to be confirmed before progressing to the next sequential step. In cases where the postconditions are not affirmed, it indicates that the desired response was not achieved, resulting in the step being deemed unsuccessful.
5. ***Substep***—refers are steps that are nested within a parent. Only one level of substep is allowed.
6. ***Logic***—set of boolean criteria and operators used to evaluate the state of the object. The outcome of this logic evaluation determines whether the step or substep can be deemed successful or unsuccessful.

Understanding context is key to assessing human reliability. Here we embedded contextual metadata within each procedure step to support the dynamic human reliability engine of HUNTER. More specifically the procedures specify Goals, operators, methods, and selection rules (GOMS)-HRA primitives for each of the sub-elements of the procedures (Boring and Rasmussen, 2016; Ulrich et al., 2017). The procedure engine is then able to take these primitives and lookup other characteristic time distribution parameters for the simulation. The procedures also specify static and dynamic performance shaping factors that influence whether the operator successfully completes the GOMS sub-elements. The end goal is to conduct several hundred of monte carlo simulations and model completion times, and success or failure of each GOMS sub-element the virtual perator engages in.

HUNTER-Rancor Implementational Modules

The implementational modules encompass the software code that facilitates the overall execution of the functional modules in HUNTER. These modules primarily function as backend code, supporting the operation of HUNTER without direct visibility to the user. While they may not be apparent to the user, these implementational modules are essential for the proper functioning and execution of HUNTER. Their purpose is to enable the smooth operation of HUNTER by providing the necessary backend functionalities required for its functioning.

Plant Modules

In addition to the implementational modules, it is important to describe the plant modules, as they play a crucial role in the overall simulation and enables several dynamic capabilities offered by HUNTER.The plant model operates in conjunction with the virtual operator's actions, keeping track of the state of the simulated plant and modeling its inherent system dynamics. It captures and monitors changes made to components by the virtual operator based on the prescribed procedure, as well as the natural progression of the simulation without any operator intervention. This includes the initiation of faults and the subsequent progression of the plant towards a system failure state. By incorporating the plant model, HUNTER is able to simulate the dynamic behavior of the plant, allowing for a realistic representation of its response to both operator actions and inherent system dynamics. The plant model enhances the fidelity and accuracy of the simulation by providing a means to track and analyze the evolving state of the simulated plant throughout the simulation process.

Scheduler Module

The Scheduler module consists of multiple classes that collaborate to conduct Monte Carlo simulations based on the tasks. **It** retains analyst-defined configurations for the entire simulation within a configuration class. This configuration class acts as a central data repository accessible throughout the simulation, ensuring that the necessary information is readily available. By storing the configurations, the scheduler facilitates efficient data management and retrieval. In addition to managing other modules, the Scheduler module encompasses various subclasses, with the log class being particularly noteworthy. The log class is responsible for generating CSV log files, enabling the output of relevant data during the simulation. By utilizing this logging functionality, the scheduler module ensures the availability of comprehensive and organized records for further analysis and review.

Task Module

During each simulation run, the Scheduler module initiates the execution of procedure steps by calling upon the Task module. The Task module comprises classes that are responsible for storing and manipulating the activities performed within each simulation run. It serves as a container for procedures and their respective steps.

By leveraging the Task module, the Scheduler coordinates the execution of procedure steps within a simulation run, ensuring the proper progression and flow of the task. This modular approach enhances the flexibility and adaptability of the simulation, enabling the simulation task to be constructed and controlled efficiently based on the defined procedures and their corresponding steps.

HRA Engine Module

One of the most important modules is the HRA (Engine) module. This is the component that models human reliability and task times. Elapsed task times are sampled from GOMS primitive time distribution parameters. Then a human error probability is calculated based on dynamic SPAR-H models (not discussed here) and the monte-carlo simulation determines the success or failure of each primitive. The virtual operator completes each step/substep until it successfully completes the task or reaches an unrecoverable terminal state.

HUNTER-RANCOR RUNS

HUNTER-RANCOR can be configured by a HRA analyst using a web tool, which generates a database of JSON files. This tool allows analysts to author procedures and assign GOMS-primitives to different elements within the procedures. Additionally, analysts can define scenarios that encompass the plant, initial plant conditions, malfunctions, operator characteristics (such as shift time), and a set of procedures to be followed.

For the purpose of development and demonstration, two scenarios were selected: "Loss of Feedwater" and "Startup from Cold-Shutdown to 100% Power." The "Loss of Feedwater" scenario is an emergency situation characterized by low complexity. In this scenario, operators are required to swiftly shut down the plant by following a series of actions in a prescribed order. On the other hand, the "Startup from Cold-Shutdown to 100% Power" scenario represents a normal operating procedure with higher complexity. It involves the coordination of various plant subsystems during the startup process.

To evaluate the performance of HUNTER, the results obtained from the simulation are compared to the human operator data collected by Chosun University in Korea, as documented in the study by Park et al. (2022). This comparison allows for an assessment of HUNTER's effectiveness and its alignment with real-world human operator behavior in the given scenarios.

Scenario 1: Loss of Feedwater

In the loss of feedwater scenario, a simple fault condition occurs where both feedwater pumps unexpectedly trip, leading to an abnormal decrease in feedwater flow. This fault occurs when the plant is online and operating at 100% power. The operators are assigned the task of following an Emergency Operating Procedure (EOP) to restore the feedwater system. However, due to the plant fault, it is not possible to restore the feedwater flow. Consequently, the operators are required to initiate and execute a rapid shutdown procedure.

To simulate this scenario, the virtual operator in HUNTER follows the EOP-0002 "Loss of Feedwater" and AOP-0001 "Rapid Shutdown" procedures. The virtual operator first verifies the loss of feedwater flow and attempts to restore it by manually restarting both feedwater pumps. However, if the feedwater flow cannot be restored, the virtual operator proceeds with rapidly shutting down the plant. This involves placing the turbine on bypass and manually tripping both the turbine and reactor.

For the purpose of analysis, HUNTER was configured to run 500 iterations of the startup scenario. The starting time-on-shift for each iteration ranges from 0 to 12 hours. The time-on-shift parameter affects the dynamic fatigue calculation within the simulation.

The loss of feedwater procedure was authored using the HUNTERweb procedure authoring interface, which provides a user-friendly environment for creating and defining procedures. Appendix A of the documentation contains a rendered version of the authored procedure, offering a comprehensive representation of the steps involved in addressing the loss of feedwater scenario.

Results

HUNTER completed the "Loss of Feedwater Procedure" and subsequent "Rapid Shutdown Procedure" in all 500 simulated evolutions, as shown in Figure 2.

Discussion

The HUNTER virtual operator completed the LOFW scenario, and the performance (timing and completion) is consistent with human operators (see Table 1). HUNTER had an average completion time of 302 seconds. The Chosun dataset had ten naïve operators and four expert operators complete the LOFW Scenario. The naïve took 195 seconds on average to complete the scenario, and the experts took an average of 2 minutes and 34 seconds to complete the scenario. We can conclude that HUNTER is slower than both naïve and expert operators when conducting abnormal operations. While this scenario is simple it demonstrates that the HUNTER/Rancor integration is functional.

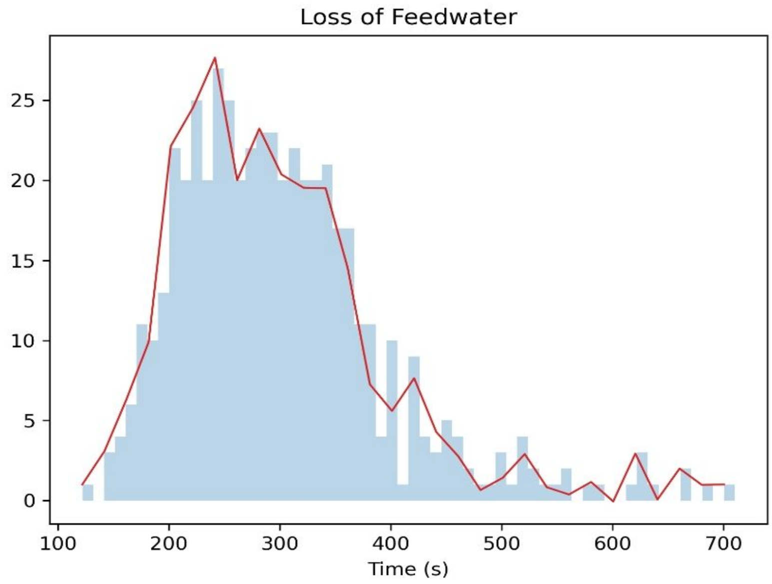

Figure 2: Task completion time distribution for virtual operator to complete loss of feedwater.

Table 1. Comparison of HUNTER and human timing for the loss of feedwater scenario.

Study	Count	Average	StdDev
Human Students	10	3:15	0:30
Human Operators	4	2:34	0:55
HUNTER Virtual	500	5:02	1:34

Scenario 2: Startup

The startup procedure in the Rancor simulation involves transitioning the reactor from a cold shutdown state to generating electricity at 100% power. This scenario is relatively complex as it requires coordination among various plant subsystems. The startup procedure for Rancor closely resembles the steps followed in a real Pressurized Water Reactor (PWR).

The startup process begins by establishing primary coolant flow and raising the control rods to initiate reactor operation. The reactor power is initially raised to a low level, typically around 10%, to facilitate the ramp-up of the turbine. Once the reactor reaches a stable state and operating temperature, the turbine can be latched, and the governor valve is raised to bring the turbine to its synchronization speed of 1800 RPM. At this point, the generator can be synchronized with the grid, resulting in the plant producing approximately 10% of its power capacity.

In the final phase of the startup procedure, the reactor power and grid load are simultaneously increased while maintaining control over reactor temperature and power levels to ensure they remain within the specified operating limits. It is crucial to avoid deviations that could trigger a reactor trip.

Although real PWRs employ more sophisticated control systems to maintain primary side temperatures and pressures, coordination is still necessary between the primary and secondary sides during load changes.

The success criterion for the startup scenario is the stable operation of the plant with the reactor operating at 100% power. This indicates that the plant has achieved the desired power generation state and is operating within the designated parameters.

In adapting the startup procedure for HUNTER, it was necessary to make the procedures more explicit and detailed compared to traditional procedures. This was done to enable the virtual operator in HUNTER to complete the steps without relying on internal knowledge of the plant and its operational controls.

However, it is important to note that the current implementation of HUNTER has a limitation in following continuous actions. Continuous actions are a mechanism in procedures that allow operators to asynchronously navigate multiple procedure steps throughout the progression of a plant evolution. In the Rancor startup procedure, human operators are responsible for monitoring and controlling reactor temperature while simultaneously carrying out the necessary steps to bring the plant online.

HUNTER was configured to run 500 iterations of the startup scenario, considering different starting time-on-shift durations ranging from 0 to 12 hours. The time-on-shift parameter has an impact on the dynamic fatigue calculation within the simulation, accounting for the operator's duration of continuous work.

By running multiple iterations of the startup scenario with varying time-on-shift durations, HUNTER can simulate a range of operational conditions and provide insights into the system's performance and potential operator fatigue issues.

Results

HUNTER completed the startup scenario in 404 of 500 iterations. Of those 404 runs it took an average time of 12 minutes 13 seconds (733 seconds). The distribution of startup times is shown in Figure 3. From the performed action across all of the scenarios we identified 8078 action attempts and a commission rate of 0.001238 (i.e., failed 10 actions). HUNTER retrieved 23,824 plant parameter indications with an error rate of 0.000839 (i.e., failed 20 checks).

Discussion

HUNTER completed the loss of feedwater scenario, and the timing and successful task completion is consistent with human operators (see Table 2) as reported in Part 1 of (Boring et al., 2022). The virtual operator had an average completion time of 302 seconds. The Chosun dataset had 10 students and 4 operators complete the Loss of Feedwater Scenario. The students took 195 seconds on average to complete the scenario, and the operators took an average of 154 seconds The virtual operator was able to successfully latch the turbine in 494 of 500 iterations and sync the turbine/generator

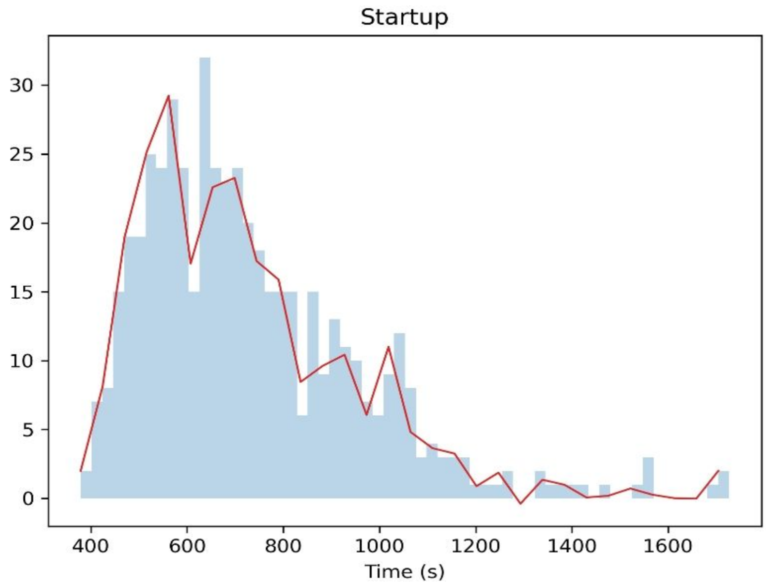

Figure 3: Distribution of times for HUNTER to complete startup.

Table 2. Comparison of HUNTER and human timing for the startup scenario.

Study	Count	Average	StdDev
Human Students	19	9:40	3:08
Human Operators	9	12:06	6:47
HUNTER Virtual	500	12:13	3:48

in 498 of 500 iterations. However, the virtual operators were only able to bring the plant to 100% power in 404 of 500 iterations. During the ramping phase, actual operators manually monitor plant parameters and increase reactor power and load. The startup procedure for the virtual operator model implemented a strategy to accomplish ramping the plant but did so less successfully than real human operators do to the slow response time for retrieving and taking actions. During this phase of the evolution the human reactor operators did not need to reference the procedures and took faster actions. The HUNTER model could be improved by adding additional GOMS-HRA primitives for modeling manual control that does not rely on formally referencing written procedures for each control decision and action.

Nineteen Chosun student participants performed the Rancor startup procedure and had comparatively more consistent and faster performance with an average time of 9 minutes and 40 seconds compared to the 12 minutes observed with the virtual operators (see Table 2). Nine operators completed the startup procedure and completed the scenario in 12 minutes and 6 seconds. The timing of the HUNTER model is consistent with the operators from the Chosun study.

The Chosun study found a task error rate of 0.009 and student operators and an error rate of 0.006 for licensed operators. The HEPs of the virtual operator were lower by an order of magnitude from the observed dataset. Future work is needed to improve the HEP modeling of HUNTER so they are representative of human operators.

CONCLUSION

The demonstration of synchronously coupling HUNTER with Rancor and successfully completing task evolutions represents a significant milestone in the application of dynamic Human Reliability Analysis (HRA). This achievement highlights the potential of integrating a virtual operator with a virtual plant model to study and analyze operator performance in real-time scenarios.

While the results are promising, it is acknowledged that further refinement is necessary in the modeling aspect to better calibrate the virtual operator's performance with that of actual operators. This refinement would enhance the accuracy and reliability of the simulation, ensuring a more realistic representation of operator behavior and decision-making processes.

The ability to debug the code from HUNTER to Rancor source code greatly facilitated the development process, particularly in dealing with the complexities of the element based procedure following. This debugging capability enhanced efficiency and helped overcome any challenges encountered during the integration process.

Furthermore, as Rancor emulates the functionality of full-scope simulators, the successful coupling achieved with HUNTER-Rancor suggests that similar integrations could be accomplished with more complex simulators. This indicates the potential for applying the HUNTER framework to various proprietary simulators and expanding its scope of application.

The observed data from the demonstrations indicate that students tend to perform faster than operators in normal evolutions such as the startup scenario. However, in abnormal evolutions like the loss of feedwater scenario, the operators exhibit faster completion times compared to the students. This suggests that operators possess the ability to adjust their work pace based on the specific circumstances of the plant, either being cautious or expedient as needed.

The timing of HUNTER closely aligns with the slower and more cautious operators in the startup procedure, indicating a good match between HUNTER and the operators' behavior in that scenario. However, HUNTER is noticeably slower in completing the loss of feedwater scenario and rapid shutdown. This suggests that the HUNTER model used here was unable to accurately capture the expediency demonstrated by the operators in such situations.

The observations made during the startup demonstration scenario indicate that while human operators engage in both procedure-guided control actions and faster freestyle manual control actions, the virtual operator in HUNTER faced challenges in effectively ramping the reactor and generator to full power. This specific phase of the evolution requires close monitoring of critical parameters to avoid tripping the reactor and turbine. The existing GOMS-HRA primitives used in HUNTER were not able to respond quickly enough in these situations.

To address this limitation, HUNTER would benefit from the inclusion of GOMS-HRA primitives that cater to both types of control actions—procedure-guided actions and faster freestyle manual control actions. The current GOMS-HRA primitives are perceived to be too slow in modeling the faster manual control actions exhibited by human operators. Even during the development of GOMS-HRA, it was recognized that refinements to the primitives would likely be necessary (Boring and Rasmussen, 2016). The startup demonstration scenario highlighted the need for such revisions.

By refining and expanding the GOMS-HRA primitives, HUNTER can better simulate the faster freestyle manual control actions performed by human operators. This would allow for a more accurate representation of operator behavior and decision-making processes, particularly in scenarios where operators need to respond quickly to critical events and make rapid adjustments.

The acknowledgement of the need for revisions to the GOMS-HRA primitives aligns with the iterative nature of developing human performance models, aiming to continually improve their fidelity and applicability. The startup demonstration scenario serves as valuable feedback to guide the refinement of the primitives and enhance the capabilities of HUNTER in capturing the full range of operator actions and responses.

To effectively incorporate continuous actions and alarm monitoring in HUNTER, additional research and empirical studies are needed. This will help in gaining a deeper understanding of operator behaviors and developing models that capture the multitasking nature of operators, their decision-making processes, and their ability to effectively manage and respond to alarms. Such advancements will contribute to the ongoing refinement and development of HUNTER, enabling it to more comprehensively simulate and analyze the performance of human operators in complex and dynamic environments.

ACKNOWLEDGEMENT

This work of authorship was prepared as an account of work sponsored by Idaho National Laboratory (under Contract DE-AC07-05ID14517), an agency of the U.S. Government. Neither the U.S. 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.

REFERENCES

Boring, R., Lew, R., Ulrich, T., & Park, J. (2022). Synchronous vs. Asynchronous Coupling in the HUNTER Dynamic Human Reliability Analysis Framework. Human Error, Reliability, Resilience, and Performance AHFE 2023.

Boring, R., Mandelli, D., Rasmussen, M., Herberger, S., Ulrich, T., Groth, K., & Smith, C. (2016). Integration of Human Reliability Analysis Models into the Simulation-Based Framework for the Risk-Informed Safety Margin Characterization Toolkit, INL/EXT-16-39015. Idaho Falls: Idaho National Laboratory.

Boring, R., Ulrich, T., Ahn, J., Heo, Y., & Park, J. (2022). Software Implementation and Demonstration of the Human Unimodel for Nuclear Technology to Enhance Reliability (HUNTER), INL/RPT-22-66564. Idaho Falls: Idaho National Laboratory.

Boring, R. L., & Rasmussen, M. (2016). GOMS-HRA: A method for treating subtasks in dynamic Human Reliability Analysis. Risk, Reliability and Safety: Innovating Theory and Practice, Proceedings of the European Safety and Reliability Conference, pp. 956–963.

Park, J., Boring, R. L., Ulrich, T. A., Yahn, T., & Kim, J. (2022). Human Unimodel for Nuclear Technology to Enhance Reliability (HUNTER) Demonstration: Part 1, Empirical Data Collection of Operational Scenarios, INL/RPT-22-69167. Idaho Falls: Idaho National Laboratory.

Ulrich, T. A. (2017). The Development and Evaluation of Attention and Situation Awareness Measures in Nuclear Process Control Using the Rancor Microworld Environment. Dissertation, University of Idaho.

Ulrich, T., Boring, R., L., Ewing, S., & Rasmussen, M. (2017). Operator timing of task level primitives for use in computation-based human reliability analysis. Advances in Intelligent Systems and Computing, 589, 41–49.