

Task and Procedure Level Primitives for Modeling Human Error

Ronald Boring¹(✉), Martin Rasmussen², Thomas Ulrich¹,
Sarah Ewing¹, and Diego Mandelli¹

¹ Idaho National Laboratory, Idaho Falls, ID, USA
{ronald.boring, thomas.ulrich, sarah.ewing,
diego.mandelli}@inl.gov

² NTNU Social Research, Trondheim, Norway
martin.rasmussen@svt.ntnu.no

Abstract. In this paper, we describe the development of behavioral primitives for use in human reliability analysis (HRA). Previously, in the GOMS-HRA method, we described the development of task level primitives, which model basic human cognition and actions. Like generic task types found in some HRA methods, the task level primitives provide a generic or nominal human error probability. These generic task types are often modeled at the task level—grouped according to a high-level goal that includes many activities. In contrast, task level primitives represent a finer level of task decomposition, corresponding not to a group of actions that comprise an overall task but rather individual steps toward that task. In this paper, we further elaborate on the task level primitives by grouping task level primitives into procedure level primitives. This terminology reflects standard groupings of activities that are performed by reactor operators when following operating procedures. For the purposes of HRA, it is desirable to model operator actions according to these prescribed procedure categories. We present mappings of the procedure level to the task level primitives found in the GOMS-HRA method. We provide examples and conclude that procedure level primitives are a useful tool to streamline HRA modeling and quantification, especially for dynamic HRA applications.

Keywords: Human reliability analysis · GOMS-HRA · Subtask modeling · Procedures · Dynamic human reliability · Analysis

1 The Importance of Subtasks in Human Reliability Analysis

In practice, human reliability analysis (HRA) looks at causes of human error at the *task* level. A task consists of a series of activities related to a common goal. This goal is typically centered on the function of a particular hardware component or system. For example, the goal to put feedwater into service at a nuclear power plant may entail multiple steps by a reactor operator, including preparatory actions like checking the availability of systems, the culmination of which is a functioning feedwater system at the plant. This division of a task into subtasks is mirrored in many operating procedure hierarchies, in which procedure steps feature substeps and in which groups of procedure steps have collective headings to indicate groupings of related steps.

The common unit of analysis in HRA is the human failure event (HFE), which is defined as the failure of a function, component, or system due to human action or inaction [1]. This definition may be seen as top-down—from the hardware to the human—in the sense that only significant hardware failures that affect the safety of the facility are considered for modeling. The human presents one of many sources of such hardware failures. As noted in [2], most human factors methods associated with task analysis tend to build a bottom-up structure of human actions—from the human to the hardware with which he or she interfaces. The bottom-up approach may capture human errors that are not readily modeled in the top-down approach, but they may not all be risk significant. Logically, therefore, it only makes sense to include human errors that actually have an impact on facility safety. However, it is possible that the top-down approach overlooks some opportunities for significant errors, such as those caused by errors of commission—actions taken by the operator that aren't required and may change the facility from its expected configuration. Moreover, the top-down approach may omit a consistent or complete modeling level of human actions, focusing instead on the most salient actions of the operator.

The risk of the HFE as a unit of analysis is that it is very high level, potentially encompassing dozens to hundreds of human subtasks related to hardware in a top-down fashion. This level of task composition is very rough and may highlight certain salient actions while overlooking seemingly less significant actions. Importantly, the HFE level is difficult to replicate consistently between analysts, as the question of what is omitted from the task list is left to analyst discretion and expertise. For this reason, inconsistent task modeling within the HFE was implicated as a significant reason for variability between human reliability analysts in one HRA benchmark [3].

The Technique for Human Error Rate Prediction (THERP) [4], arguably the original HRA method, did not explicitly model HFEs. Instead, it considered groups of human actions within an HRA event tree, a unique representation for linking subtasks. This structure was particularly important for THERP, because it specified how the human error probability (HEP) should be calculated. Each node in a THERP HRA event tree is a human subtask; the tree models how HEPs, recoveries, and dependence between subtasks occur. In most cases, the HRA event tree as a whole may be seen as an HFE, and the total HEP is associated with the interconnections between subtasks. The subtasks are quantified individually through lookup tables, which create a type of scenario-matching approach. Novel subtasks are mapped to similar subtasks in THERP tables, for which HEPs are provided. Dependence in the propensity of error to beget error, resulting in increased error rates for the second and subsequent subtasks in a series. In contrast, recovery breaks the error chain and puts the operator back on a success path. Whereas dependence increases the HEP, recovery decreases it.

It can be extremely labor intensive to complete an HRA in THERP, in large part due to the necessity to model many subtasks. One of the chief simplifications of later HRA methods was the focus on the HFE instead of the subtask level of analysis. While no HRA method to our knowledge specifically advocates omission of a thorough task analysis, there is nonetheless an erosion of such efforts in common practice, because most probabilistic risk models only require input of a single HEP for the entire HFE. Even THERP and methods directly derived from it (e.g., [5]) currently tend to be used primarily for quantification of overall HFEs, not subtask quantification.

While there appears to be a trend away from subtask modeling in HRA as practiced, subtask modeling remains especially important for dynamic HRA, in which human activities are placed into a simulation. For example, in a recent dynamic HRA effort called the Human Unimodel for Nuclear Technology to Enhance Reliability (HUNTER) [6], a virtual human operator is modeled alongside thermohydraulic code. HUNTER models each required human intervention in the plant with corresponding plant response. HUNTER actions are therefore being modeled for each step in the operating procedures, autocalculating the HEP [7] and required completion time for each step. While dynamic HRA may be considered creating a virtual operator, it is also possible to conceive of this type of modeling as creating a virtual analyst whose job it is to calculate the HEP automatically for each HFE based on available information in the model [8]. Regardless of whether it is a virtual operator or analyst, dynamic HRA requires the subtask level of granularity when considering human actions. In this paper, we focus on subtask modeling to accommodate current efforts at modeling operator actions in the HUNTER dynamic HRA framework.

2 GOMS-HRA for Subtask Modeling

The Goals-Operators-Methods-Selection rules (GOMS) approach is a method developed to factor human information processing (i.e., cognition) into empirical observations [9]. GOMS is an important method for decomposing human activities into their constituent subparts. It may be considered a task analysis approach focused on the subtask level of mental operations. Adaptations of GOMS, like the Keystroke Level Model (KLM) [10] have been used to assign action specific timing data to human subtasks. Combining these subtask durations allows analysts to determine the relative efficiencies of different system designs, for example.

GOMS was recently adapted to HRA to create GOMS-HRA [11] to encompass the subtask level of human activities for dynamic HRA. Because GOMS is a framework more than a specific method for considering human activities at the subtask level, GOMS-HRA pays homage to GOMS, but it should not be considered an adaptation of any of the previous instantiations of the method.

GOMS-HRA is linked to a taxonomy of human subtask primitives, corresponding to basic human activities likely to be performed in conjunction with operating a nuclear power plant. The Systematic Human Error Reduction and Prediction Approach (SHERPA) [12] serves as the basic taxonomy, with adaptations for cognitive decision making [13] and periods of relative inactivity such as waiting and monitoring. The modified SHERPA taxonomy for use in GOMS-HRA is provided in Table 1. In our nomenclature, we call this list *task level primitives* (TLPs).

Note that the action (*A*), checking (*C*), retrieving (*R*), and selecting (*S*) TLPs make a distinction between control boards (i.e., main control room) and field (i.e., balance of plant) operations, as denoted by a subscripted $_C$ or $_F$, respectively. The instruction (*I*) TLP distinguishes between producing ($_P$) and receiving ($_R$), respectively. The decision making (*D*) TLP considers decisions guided by procedures ($_P$) or without procedures ($_W$).

Table 1. GOMS Operators used to define Task Level Primitives.

Primitive	Description
A_C	Performing required physical actions on the control boards
A_F	Performing required physical actions in the field
C_C	Looking for required information on the control boards
C_F	Looking for required information in the field
R_C	Obtaining required information on the control boards
R_F	Obtaining required information in the field
I_P	Producing verbal or written instructions
I_R	Receiving verbal or written instructions
S_S	Selecting or setting a value on the control boards
S_F	Selecting or setting a value in the field
D_P	Making a decision based on procedures
D_W	Making a decision without available procedures
W	Waiting

Note also that the difference between checking (C) and retrieving (R) has to do with the level of information being sought. Checking is the simple act of confirming information like a status light on the control boards. In contrast, retrieving requires greater cognitive engagement such as reading the exact value on a gauge or trending a value over time. Generally speaking, the more cognitive effort that is required, the more the categorization falls to retrieval.

This taxonomy serves not only to decompose human activities into elemental subtasks; the taxonomy also affords the ability to anticipate common error types for each subtask. The adapted SHERPA taxonomy from Table 1 yields the following types of errors at a high level:

- *Action Errors*—Performing the required action incorrectly or failing to perform the action
- *Checking Errors*—Looking for required information in wrong place or failing to look for that information
- *Retrieval Errors*—Obtaining wrong information such as from control room indicators or failing to obtain required information
- *Information Communication Errors*—Communicating incorrectly or misunderstanding communications
- *Selection Errors*—Selecting the wrong value or failing to select a value
- *Decision Errors*—Making wrong decision or failing to make decision.

Waiting is not a TLP in the sense of modeling failed actions and HEPs; instead, it acts as a placeholder for tasks such as monitoring that involve extended periods of time. Therefore, waiting is not modeled as a potential error type, although we acknowledge that there is opportunity for human errors to occur while waiting. Each of the errors associated with TLPs can, in turn, be decomposed into further error types similar to what is found in [14]. GOMS-HRA stops short of providing a catalog of possible

failure mechanisms for each TLP, although such a catalog may be the topic of future research efforts.

As noted, HRA methods like THERP use a scenario-matching approach for quantification. The task or subtask at hand is compared against a lookup table of prequantified nominal HEPs and subsequently fine-tuned through further analysis. Similarly, the TLPs can serve as a series of generic task types with associated nominal HEPs. Table 2 includes nominal HEPs for each of the TLPs, as aligned to THERP subtasks [4] using expert judgement. Unlike THERP, which includes fairly specific scenario matches, the GOMS-HRA TLPs are characterized by the type of human activity rather than a specific scenario. As such, we believe the TLPs are more generalizable than the scenarios found in THERP. The TLPs allow maximum flexibility for modeling human activities in a dynamic simulation.

Table 2. HEPs associated with each talk level primitive.

Operator	Nominal HEP	THERP source ^a	Notes
A_C	0.001	20-12 (3)	Assume well-delineated controls
A_F	0.008	20-13 (4)	Assume series of controls
C_C	0.001	20-9 (3)	Assume well-delineated indicators
C_F	0.01	20-14 (4)	Assume unclear indication
R_C	0.001	20-9 (3)	Assume well-delineated indicators
R_F	0.01	20-14 (4)	Assume unclear indication
I_P	0.003	20-5 (1)	Assume omit a step
I_R	0.001	20-8 (1)	Assume recall one item
S_C	0.001	20-12 (9)	Assume rotary style control
S_F	0.008	20-13 (4)	Assume series of controls
D_P	0.001	20-3 (4)	Assume 30-minute rule
D_W	0.01	20-1 (4)	Assume 30-minute rule
W	n/a	n/a	n/a

^a Corresponds to THERP [4] Table values from Chap. 20.

3 Introducing Procedure Level Primitives

The TLPs are at a more basic level than are the actions commonly prescribed to reactor operators. Reactor operators follow operating procedures ranging from standard operating procedures, annunciator response procedures, abnormal operating procedures, emergency operating procedures, to severe accident management guidelines (SAMGs). SAMGs tend to be different from the rest of the procedures in that they provide problem solving strategies rather than step-by-step processes. The remaining procedures articulate step-by-step actions the operators should follow to maintain production and safety at the plant. In fact, there are often license penalties for deviating from the procedures, making them legally binding process manuals.

Because the procedure steps serve as the script for operating the plant, the logical level of task decomposition for HRA is the procedure step. Procedure steps and

substeps explicate exactly what the reactor operators need to be doing at the plant, including any interfacing with components, and in which sequence. Procedures also point to specific decision branch points. As such, it is possible to create a simplified model of the reactor operator without necessarily creating a full-blown artificial intelligence system [15]. The level of procedural detail and the high procedural compliance create a perfect context for using procedures for simplified dynamic modeling frameworks like HUNTER.

To link the TLPs to the procedures, we developed *procedure level primitives* (PLPs) that map common procedure steps to TLPs. In many cases, a single procedure step may actually entail a series of TLPs. Consider, for example, the common procedure step to check something such as an indicator (see Fig. 1). This procedure step corresponds to the TLPs of check (C_C), making a decision based on procedures (D_P), verbalizing the value to the shift supervisor (I_P), selecting or setting a value (S_C or A_C) if necessary, and potentially waiting (W) while monitoring the value.

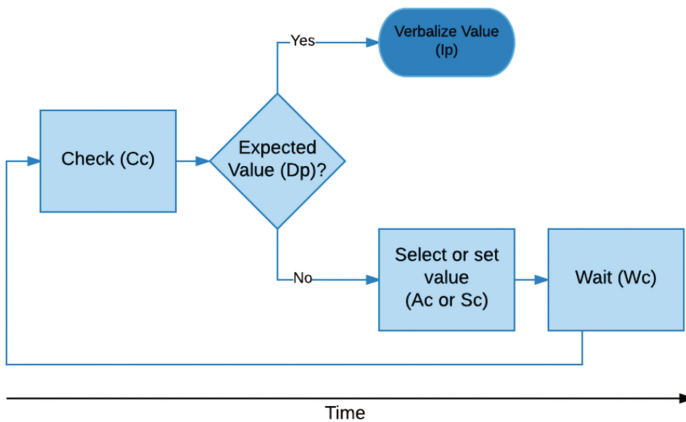


Fig. 1. Procedure level primitive decomposition into task level primitive example.

To simplify the process of modeling TLPs, we have mapped a number of common procedure steps to TLPs (see Table 3). These mappings, which constitute the PLPs, may be reused across analyses and may make it possible to extract TLPs in an automated fashion from operating procedures.

To arrive at a standard list of PLPs, we referenced the Procedure Professionals Association (PPA) *Procedure Writer's Manual* [16]. The PPA manual provides an extensive list of action verbs and their definitions to guide procedure development at nuclear power plants and other facilities. An example definition for check is:

CHECK: Observe an expected condition exists
(no actions to correct)

Table 3. Common procedure level primitives mapped to task level primitives.

Procedure level primitive	Task level primitive	Mapping Notes
Determine*	C_C or R_C	Information type dependent
Ensure*	C_C or R_C and/or A_C and/or S_C	Information and control action type dependent
Initiate	A_C	–
Isolate	A_C	–
Minimize	S_C	–
Open	A_C	–
Verify*	C_C or R_C	Information type dependent
Check*	D_P , A_C , S_C , W_C , and/or I_P	Information type dependent

* These procedure level primitives, or action verbs, can be decomposed into multiple task level primitives. Figure 1 depicts the check procedure primitive decomposed into D_P , A_C , S_C , W_C , and I_P task level primitives and the relationship between these task level primitives.

The list of procedure steps (or action verbs) provided by PPA is extensive, but it is not necessarily exhaustive of all possible procedure steps at plants, nor does it narrow the list of procedure steps according to frequency. Moreover, the PPA manual is only a guideline, meaning individual plant’s use of particular preferred procedure terminology or adherence to the suggested definitions will vary. Indeed, the consistency between procedures within individual plants varies, depending on the procedure writers and operators involved in generating the procedures as well as the system being proceduralized. There can be, for example, considerable differences in preferred nomenclature between procedures for the main control room vs. field operations. Mapping all procedure steps in the PPA list as PLPs would prove difficult and be fraught with necessary subjective interpretations to meet plant specific details. Instead of a complete mapping, we have mapped PLPs on a case-by-case basis as we have encountered new steps in procedures we are modeling as scenarios in HUNTER. This process, over time, is building a library of PLPs. Common PLPs that recur across many procedures are found in Table 3.

4 Discussion

4.1 Complex Mappings

As demonstrated with the check procedure step, PLPs can consist of multiple TLPs. The challenge with the reuse of PLPs across analyses is that complex mappings may not always be consistent. In one case, a check step may be synonymous with the checking (C_C) TLP, but, in another case, it may require multiple TLPs. Where complex mappings occur, model building will still require expertise by the human analyst to ensure the mappings are appropriate. There is still value in reusing PLPs, since the

suggested mappings can serve as a template for reuse, but complex PLPs will likely require manual fine-tuning to suit their use context.

Using text mining, we have explored the possibility to extract procedure steps to derive PLPs automatically from operating procedures [17]. This process is still in the early stages of exploration. One of the main challenges of text mining PLPs is with complex PLPs, whereby the exact mappings to a series of TLPs requires more contextual information than can readily be extracted automatically. Other challenges remain. The formatting of procedures (e.g., the common double column format for procedures for Westinghouse pressurized water reactors) presents puzzling logic and parsing for current text mining algorithms. Certain placeholder words like `if` and `not` are at least as meaningful as standard action verbs, yet these words are semantically excluded from text mining. Finally, there are many challenges in differentiating what actions a reactor operator is doing versus what the plant is doing. For example, consider the following illustrative procedure step:

```
check that valve is closed and pressure is decreasing
```

The operator action is `check`. However, there are potential points of confusion over the related word stems of `close` and `decrease`. There remains considerable development work to refine the automatic extraction of PLPs from operating procedures. Currently, we are manually extracting the procedure steps and logic to arrive at accurate models of operator actions.

4.2 The Problem with Procedures

Of course, one of the main limitations of the PLP approach is that it relies on operating procedures. A few specific limitations are detailed below:

- *Variability in procedures.* Terminology varies considerably between plants, but there may also be intra-plant variability between procedures depending on the system addressed. Such variability is not a limitation of the procedures or the plant, but it makes the process of creating a global set of PLPs implausible. To address this reality, we have crafted PLPs on a procedure-by-procedure basis, tailoring the underlying TLPs as needed. In our anecdotal experience, the PLPs have proved robust in mapping TLPs. Initial experience suggests strong suitability to reuse of the PLPs across procedures, although we are carefully vetting individual instantiations of them to account for variability across procedures.
- *Procedures as scripts.* Procedures do not represent the totality of operator actions. There are routine tasks such as monitoring and communicating that are so frequent as not to warrant mention in specific procedures. While it is tempting to use the procedures as a script for modeling operator actions in HRA, doing so could result in an unrealistic account of what operators do. It would omit key routine activities, but it would also suggest a linearity of actions that is not representative of operator actions. Operators sometimes need to deviate from procedural paths or select alternate procedures in response to emerging plant conditions. While procedures may be a good starting point for modeling operator tasks, they actually underspecify

everything the reactor operators are doing. At best, procedures can be used to create simplified models of operator actions.¹ One key advantage of PLPs is that they can incorporate some of the underspecified actions—the between-the-lines activities of the operators—as part of the PLP. If, for example, each operator action is assumed to include some form of threeway communication between crew members, the PLPs for each procedure step should include the instructions (*I*) TLP. PLPs can ensure that the right TLPs are included in the analysis.

- *Task roles.* Procedures generally provide good detail about what tasking needs to be completed, but they do not specify who is completing that tasking. This is not an omission in the procedures; rather, it is by design, because the assignment of specific tasks or procedure steps is the job of the control room supervisor (CRS). The CRS maintains situation awareness of plant conditions and processes, follows and often reads aloud procedures, and delegates tasking between available crew members. The net effect whether the operator at the controls or the balance-of-plant operator performs a particular procedural step is negligible in most HRA models. However, for dynamic HRA, where the goal is to create a virtual operator, crew roles do matter, especially for determining relative workload of specific operators. As such, procedures cannot be used blindly, but rather must be augmented to specify which operator is performing the tasks.
- *Complex logic and compound steps.* As discussed briefly in the context of text mining, procedures are rarely as simple as one step equals one action. Instead, procedures often feature complex AND/OR logic and compound steps. Complex logic can be accounted for with TLPs related to decision making, and compound steps are simply chains of TLPs. The PLPs can likewise track the TLPs for added complexity and steps, but the generalizability of such PLPs may prove minimal for complex steps.

It is telling that many HRA methods include two considerations of procedures as a performance shaping factor. The first aspect is the *procedural quality for the task*. Procedure writers strive to cover plant conditions as completely as possible, but it is never possible to anticipate every process permutation at the plant. Compound faults, for example, may force operators to prioritize their response and choose between competing procedures. The second aspect of procedures considered in many HRA methods is *procedural adherence*. Reactor operators are trained both to follow operating procedures and to recognize when procedures may not adequately cover the breadth of possible responses. The reactor operators must exercise their expertise, which may on occasion take them outside of the procedural script or branch them to a new procedure that is a better fit to the plant conditions. There are multiple success paths to recover from a plant upset, for example, and crews may respond differently throughout the evolution of the transient.

Since the procedural quality and procedural adherence are known to vary, these will certainly limit the ability of the PLPs to become one size fits all across analyses. Of

¹ Note that a simplified model based on PLPs will likely be incrementally more detailed than a simplified HFE-level model. One premise of GOMS-HRA is that the more detail that is available in the HRA model, the higher the fidelity and scrutability of the quantification.

course, the purpose of PLPs is not to be generic constructs that are interchangeable across all contexts. The PLPs are simply a way of bundling common groups of TLPs that are associated with procedure steps. If the procedure steps do not lend themselves to PLPs, a more detailed case-by-case TLP analysis is warranted.

4.3 Advantages of PLPs

While the preceding discussion has highlighted some challenges and shortcomings of using PLPs, we believe there is merit in the approach. Where appropriate, PLPs give a way of linking procedure steps to TLPs in GOMS-HRA. This approach can greatly benefit efforts at dynamic HRA modeling in frameworks like HUNTER by providing the basis for error quantification and even task timing [18]. Additionally, the TLPs provide a basis for anticipating certain types of errors that might occur in the context of procedure following. In many cases, PLPs can be reused, thereby reducing the laborious efforts associated with model building. Ultimately, the PLP approach provides a consistent way to decompose procedure steps into meaningful subtasks in HRA. This approach is especially useful for dynamic HRA for heavily proceduralized nuclear power plant activities, but PLPs hold equal promise for any HRA that requires subtask modeling.

Acknowledgements. This work of authorship was prepared as an account of work sponsored by 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.

Hoomaikai Hawai'i, no ka le'ale'a ka hihia oiai kākau i kēia pepa. The authors express their thanks to Dr. Curtis Smith at Idaho National Laboratory for championing the research of HUNTER and GOMS-HRA and funding various parts under the U.S. Department of Energy's Light Water Reactor Sustainability program and Idaho National Laboratory's Laboratory Directed Research and Development program.

References

1. American Society of Mechanical Engineers: Addenda to ASME/ANS RA-S-2008 Standard for Level 1/Large Early Release Frequency Probabilistic Risk Assessment for Nuclear Power Plant Applications, ASME/ANS RA-Sb-2013, New York (2013)
2. Boring, R.L.: Top-down and bottom-up definitions of human failure events in human reliability analysis. In: Proceedings of the Human Factors and Ergonomics Society 58th Annual Meeting, pp. 563–567 (2014)
3. Poucet, A.: Human Factors Reliability Benchmark Exercise, Synthesis Report, EUR 12222 EN. Office for Official Publications of the European Communities, Luxembourg (1989)

4. Swain, A.D., Guttman, H.E.: Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications, NUREG/CR-1278. U.S. Nuclear Regulatory Commission, Washington, DC (1983)
5. Gertman, D., Blackman, H., Marble, J., Byers, J., Smith, C., O'Reilly, P.: The SPAR-H Human Reliability Analysis Method, NUREG/CR-6883. U.S. Nuclear Regulatory Commission, Washington, DC (2005)
6. Boring, R., Mandelli, D., Rasmussen, M., Herberger, S., Ulrich, T., Groth, K., Smith, C.: Human unimodel for nuclear technology to enhance reliability (HUNTER): a framework for computational-based human reliability analysis. In: 13th International Conference on Probabilistic Safety Assessment and Management (PSAM 2013), paper A-531 (2016)
7. Rasmussen, M., Boring, R.L.: Implementation of complexity in computation-based human reliability analysis. In: Proceedings of the European Safety and Reliability Conference on Risk, Reliability and Safety: Innovating Theory and Practice, pp. 972–977 (2016)
8. Rasmussen, M., Boring, R., Ulrich, T., Ewing, S.: The virtual human reliability analyst. In: Proceedings of the 1st International Conference on Human Error, Reliability, Resilience, and Performance, Los Angeles (2017, in press)
9. Card, S.K., Newell, A., Moran, T.P.: The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, Hillsdale (1983)
10. Card, S.K., Moran, T.P., Newell, A.: The keystroke-level model for user performance time with interactive systems. *Commun. ACM* **23**(7), 396–410 (1980)
11. Boring, R.L., Rasmussen, M.: GOMS-HRA: a method for treating subtasks in dynamic human reliability analysis. *risk, reliability and safety: innovating theory and practice*. In: Proceedings of the European Safety and Reliability Conference, pp. 956–963 (2016)
12. Stanton, N.A., Salmon, P.M., Rafferty, L.A., Walker, G.H., Baber, C.: *Human Factors Methods: A Practical Guide for Engineering and Design*, 2nd edn. Ashgate Publishing Co., Aldershot (2013)
13. Boring, R.L.: Defining human failure events for petroleum applications of human reliability analysis. *Procedia Manufact.* **3**, 1335–1342 (2015)
14. Whaley, A.M., Xing, J., Boring, R.L., Hendrickson, S.M.L., Joe, J.C., Le Blanc, K.L., Morrow, S.L.: *Cognitive Basis for Human Reliability Analysis*, NUREG-2114. U.S. Nuclear Regulatory Commission, Washington, DC (2016)
15. Coyne, K.A.: *A Predictive Model of Nuclear Power Plant Crew Decision-Making and Performance in a Dynamic Simulation Environment*. PhD Dissertation, University of Maryland, College Park (2009)
16. Procedure Professionals Association: PPA AP-907-005, Procedure Writer's Manual, Revision 2 (2016)
17. Ewing, S.M., Boring, R.L., Rasmussen, M., Ulrich, T.: Text mining for procedure-level primitives in human reliability analysis. In: Proceedings of the 1st International Conference on Human Error, Reliability, Resilience, and Performance, Los Angeles (2017, in press)
18. Ulrich, T., Boring, R., Ewing, S., Rasmussen, M., Mandelli, D.: Operator timing of task level primitives for use in computation-based human reliability analysis. In: Proceedings of the 1st International Conference on Human Error, Reliability, Resilience, and Performance, Los Angeles (2017, in press)