Refining Supervisory Control Capability for Target User Populations

Robert E. Wray(✉), Randolph Jones, Charles Newton, and Ben Bachelor

Soar Technology, Inc., 3600 Green Court Suite 600, Ann Arbor, MI 48105, USA
{wray, rjones, charles.newton, ben.bachelor}@soartech.com

Abstract. In designing and developing intelligent automation systems, there is often tension between the computational capabilities of the automated system and its usability and understandability. This paper presents a case study in which this tension was manifest and how we attempted to resolve it in a particular application. The application requires intelligent automation in distributed simulation for training. We describe an initial approach to such control, feedback we received from potential users, and a revision to the capability that eliminated some features but that was more acceptable to the user community. This case study may offer some observations and lessons applicable to other domains, especially in situations where penetration of a technology into everyday use is driven by informal user adoption criteria (transparency, trust, perceived usability) alongside formal functional requirements.

Keywords: Supervisory control · Simulation-based training · Requirements analysis and design · Interaction design

1 Introduction

Automation technologies are playing larger and more important roles in the lives of much of world's population. Aircraft largely fly themselves; soon automobiles may also be controlled in the main by automation. Industrial systems and processes are also becoming increasingly automated. Via networks, cloud-based data, and every-present controllers (like a smartphone), everyday tasks in the industrialized countries also are becoming more automated: computer assistants find information, schedule meetings and resolve conflicts, etc.

In designing and developing automation systems, there can be tension between the computational capabilities of an automated system and its usability and understandability by users [2, 3]. This tension is especially important for supervisory control applications, where a user sets conditions and preferences for the goals and performance of the automation and then the automation attempts to carry out its tasks, according to those conditions, which the user can adjust, override, or change during execution [4, 5]. In the examples introduced above, although the conditions for performance and the frequency of interaction differ, they all require the user's on-going

M. Kurosu (Ed.): HCI 2017, Part II, LNCS 10272, pp. 721–731, 2017.
DOI: 10.1007/978-3-319-58077-7_56

interactions with the automation system to carry out tasks successfully, at least from the point of view of the user.

From a human-machine systems perspective, the automation system fulfills its purpose only to the extent it automates the execution of some task (flying the aircraft) and that it does so in the service of its users goal (e.g., staying within standard flight lanes, maintain a smooth flight while minimizing fuel use, etc.). In other words, the application of automation does not necessarily lead to increased task effectiveness. Users may both overly rely on automation and fail to use ("disuse") automation what it could support improved task performance [2]. Models and empirical studies provide useful design guidance [5–7] but finding a resolution to this tension for some new, specific application is seen as more art than science [8].

This paper presents a case study of a manifestation of this tension and how we attempted to resolve it in a particular application. The application requires supervisory control in distributed simulation for training. Below, we describe the problem that the automation was seeking to address and then introduce an initial approach, which we term *Dynamic Tailoring,* that addressed the functional requirements successfully. However, as we provided demonstrations and prototypes to potential users, we discovered that some of the functional requirements our analysis had identified were, in practice, less of a priority than the user's ability to customize, understand and trust the automated systems.

In response, we developed a significantly different capability that eliminated some features but that privileged usage requirements over some of the original functional requirements. This second capability, which we call the Scenario Director, was more acceptable to the user community and led to adoption, even though its ability to adapt scenarios is both less autonomous and less informed by learner state than the original Dynamic Tailoring prototype. The case study may offer some relevant observations and lessons applicable to other domains, especially in situations where penetration of the technology into everyday use is driven as much by informal user adoption criteria than formal functional requirements.

2 Dynamic Adaptation of Simulation Scenarios

Simulation offers a number of advantages over live training, especially for domains where repeated practice is expensive. When simulation-based training requires the presence of actors other than the trainee, simulated role-players (constructive entities, semi-automated forces, human behavior models, etc.) often are deployed to provide a reasonably realistic simulation of the interactive environments in which training occurs. These simulated role-players are typically designed to provide realistic but common/routine behavior within the context of a training simulation. For example, in tactical aviation training, a simulated "red" entity might present realistic tactics that a "blue" human pilot might expect to see in some real-world setting.

It is common, however, for a training simulation or exercise to require more of simulated role-players than typical or routine behaviors. Table 1, summarizing a more thorough analysis [9], lists situations where typical behaviors fall short of meeting the full spectrum of requirements. Typically, this gap leads to one of two outcomes:

Table 1. Scenario requirements not met by typical approaches to simulated role-players.

Requirement		Description
1	Narrative coherence	A trainee's decisions and actions in the scenario can result in a situation where the pre-defined goals of simulated role-players are no longer rational (e.g., positioning a tanker to refuel aircraft that have been inadvertently eliminated). A scenario should adapt to ensure that the future events within a scenario are consistent with past ones [10]
2	Target training goals	The trainee's behavior early in the training scenario can lead to a situation where a specific training opportunity designed for the training scenario cannot be satisfied by simulated role-players. A scenario should ensure that the trainee can undertake practice for specific training goals
3	Adapt to trainee skill and proficiency	A training scenario can be poorly matched to an individual trainee's current level of skill, potentially resulting in frustration (too hard) or boredom (too easy). Scenarios should attempt to meet (or slight exceed) each trainee's current level of capability (i.e., the "zone of proximal development" [11])
4	Conform to scenario and range constraints	In many cases, non-tactical behavioral constraints may be important for a scenario (e.g., rules for maintaining safety of flight in an LVC scenario). Scenarios should be able to enforce these constraints on participants, including simulated role-players

1. **Introduction of human intervention and control:** Human simulation operators are used to control and to direct the low-level behaviors of simulated role-players. This approach greatly increases the cost of simulation-based training, mitigating some of the advantage of using simulation rather than live training.
2. **Reduced training effectiveness:** In the absence of human control, training simulation may introduce unrealistic behavior (and potentially negative training) and may be inefficient (the resulting simulation does not provide the desired training opportunity). This approach also introduces additional costs at the systems level, manifest in either increased training (greater time on task) or reduced operational capability (trainees graduating with reduced average skill).

The requirements in Table 1 are requirements for a training scenario that is presented, rather than a requirement for any individual role-player in that scenario. We use the term *dynamic scenario adaptation* to refer to technical approaches for meeting such requirements. As summarized in [9], there are many architectural approaches and algorithms that could be used to realize dynamic scenario adaptation. In the next section, we introduce an initial approach that attempted to address all these requirements.

3 Trainee-Focused, Autonomous Scenario Adaptation

Having identified the requirements outlined in Table 1, we developed a prototype capability designed to address them. The high-level architectural approach is summarized in Fig. 1. This design pattern derives from the use of director agents in

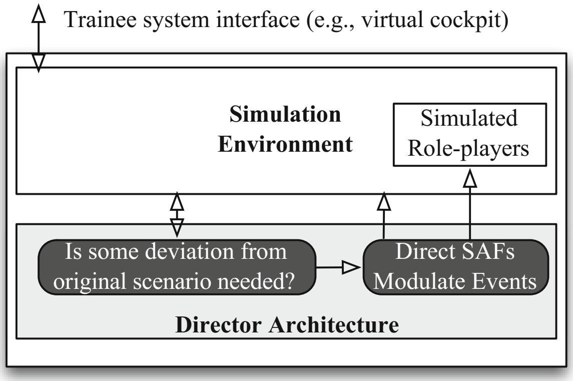
Trainee system interface (e.g., virtual cockpit)
Simulation Environment
Simulated Role-players
Is some deviation from original scenario needed?
Direct SAFs Modulate Events
Director Architecture

Fig. 1. The director architecture design pattern.

computer games [12, 13] and is here termed the *Director Architecture*. There are two primary components, a recognition component that identifies when and what change in the scenario is needed and then a direction and control component that interfaces to the simulation to modulate events. Because simulated role-players are such a critical component of these systems, the design pattern calls out the specific need in a functional system to control and to direct these role-players. The target result is a software capability that functionally acts similarly to the way an operator acts, taking control of a simulated role-player and over-riding its "native" behavior. One advantage of this approach, over different architectural approaches, is that it does not require significant changes to existing (and sometimes validated) libraries of behaviors of simulated role-players [9].

We had previously developed a software capability that instantiated this director design pattern with a particular focus on adaptation based on training need and trainee capability [14]. This *Dynamic Tailoring System* embeds a learner proficiency model to adapt and to customize scenario based on observer learner skill [15] and includes an explicit representation of a training scenario narrative [10] to enable adaptation for narrative coherence. Because training scenarios unfold in real time and instructors in many domains interact with students within the training context, the Dynamic Tailoring System is intended to be largely autonomous during scenario execution; instructors configure the system for the scenario and encode preferences for choices and then the system attempts to satisfy those preferences during scenario execution without requiring or relying on instructor control during execution.

We applied this capability to the tactical air training domain and showed it could be used to adapt realistic training scenarios to the requirements summarized in Table 1 [1]. Figure 2 illustrates the software implementation of the prototype. The Monitor and Pedagogical Manager components realize the recognition of when adaptation is needed and the Experience Manager undertakes actions in response to recognition, including both direct manipulations of events in the simulation as well as directing individual entities. The algorithms within these components depend on a number of data stores, most notably an assessment model (designed to determine how "correct" a trainee's actions are) and a proficiency model to track performance over time.

Given the potential complexity of the Fig. 2 system, it was designed to be configured and controlled by users via a series of high-level scenario-adaptation settings. For example, potential action choices of the Experience Manager could be labeled according to how directly helpful an action was, how simple or complex it was, and if

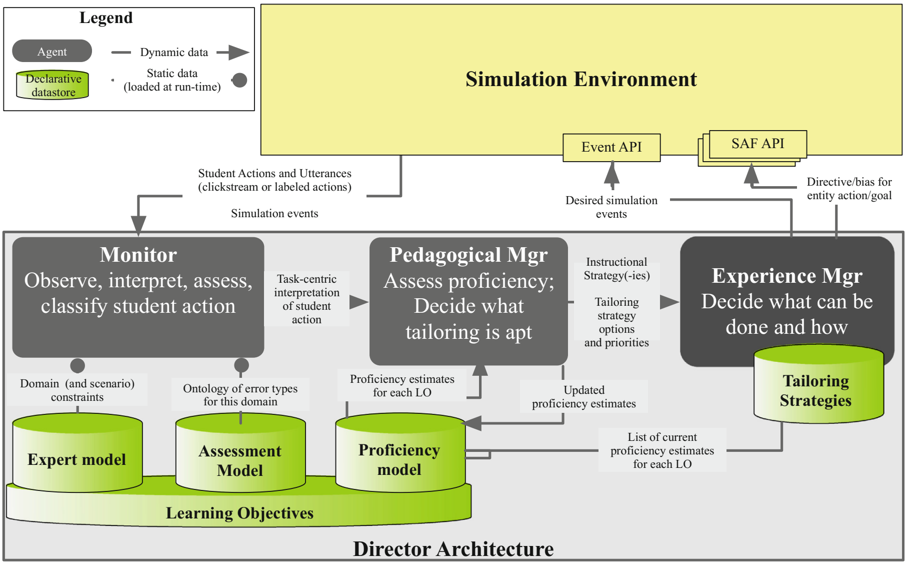
Legend
Agent
Dynamic data
Declarative datastore
Static data (loaded at run-time)
Simulation Environment
Event API
SAF API
Student Actions and Utterances (clickstream or labeled actions)
Simulation events
Desired simulation events
Directive/bias for entity action/goal
Monitor
Observe, interpret, assess, classify student action
Task-centric interpretation of student action
Pedagogical Mgr
Assess proficiency; Decide what tailoring is apt
Instructional Strategy(-ies)
Tailoring strategy options and priorities
Experience Mgr
Decide what can be done and how
Tailoring Strategies
Domain (and scenario) constraints
Ontology of error types for this domain
Proficiency estimates for each LO
Updated proficiency estimates
List of current proficiency estimates for each LO
Expert model
Assessment Model
Proficiency model
Learning Objectives
Director Architecture

Fig. 2. Dynamic tailoring software design (adapted from [1]).

reflected a frequent/typical action or observation in the environment or an atypical/infrequent one. The motivation for this labeling was to enable the system to make choices according to an instructor's tailoring preferences without requiring an instructor to specify and to configure individual tailoring choices. In subsequent work in another domain, with different instructional goals and requirements, this approach was shown to enable instructors and other representative users to customize and configure a training scenario in about a day of effort [16].

4 Refining Requirements Based on User Feedback

In the tactical-air training domain, however, this approach, while functionally sufficient, introduced a number of concerns in the user community. Table 2 summarizes feedback we received as we demonstrated the Dynamic Tailoring prototype to potential users and simulation-system stakeholders. These comments suggest that potential users were concerned with both a lack of transparency [7] and usability [17]. Because we were not able to conduct formal and sustained user testing, it is not clear if greater familiarity would have resulted in the development of greater trust with experience, which is sometimes the case [18]. Generally, the users acknowledged that the scenario adaptation capability fulfilled a need but that the complexity of the solution introduced new requirements on users that would have made it difficult and rare to use (and meet the scenario adaptation need) in practice.

Most importantly, in this domain, a simulation operator is typically present during training to customize the training experience for the trainee, based on guidance from the instructor. Thus, to a significant extent, the limitation of the initial prototype was its

Table 2. Feedback and recommendations on the initial scenario-adaptation prototype

Category	Description
Tailored quality of presentation	Users desired that the *quality of presentation* should be customized for individual trainees based on subjective assessment of current actions. Most importantly, users resisted the concept of adaptation based on a proficiency model, because a pilot's tactical proficiency includes subjective assessments that cannot be readily codified (e.g., two students with similar observed competencies might receive different real-time interventions based on instructor judgment alone)
Direct behavioral responses	Users expressed a preference for direct specification of what behavioral responses should be presented in specific situations. They persisted in this preference even though many potential users are not familiar with behavior specification tools and development
Instructor-controlled and predictable responses	Instructors do not want to be surprised during the course of scenario execution. They expressed preference for direct control of when adaptation should occur, rather than adaptation based on narrative coherence or learner skill. They want to control when interventions are initiated and what interventions are introduced
New workflows	Users were concerned about having to learn to use new tools and develop new processes and workflows to support them. This concern was both related to having to codify information (such as proficiency) that is subjectively assessed in the current culture as well as the increased time to configure the components to enable an adaptive scenario

assumption that largely autonomous adaptation was needed for this domain. Instead, instructors desired adaptation to be driven by simulation operators and to focus on reducing their workload and enabling control of simulated role-players at a more abstract and task-focused level than they were able to accomplish with existing technology. For example, in a situation where the scenario requires that separate groups of aircraft maintain a desired distance from one another, the current level of control requires manually directing the heading and speed of aircraft to maintain the separation. Higher-level control might enable an operator to specify a minimum separation distance and require the intelligent automation to carry out how to satisfy this requirement as the scenario unfolds.

The feedback helped us understand that the existing training environments targeted for this technology were culturally and technically better suited for a supervisory control solution than an autonomous one.

5 Scenario-Focused Supervisory Control of Scenario Adaptation

Based on the feedback received for the initial prototype, we reviewed the initial requirements and reconsidered both the architectural approach (Fig. 1) and the detailed technical design (Fig. 2) for scenario adaptation. In reviewing the initial requirements, we recognized that all the requirements (i.e., narrative coherence, targeted training goals, proficiency-based adaptation, and conformance to exercise constraints) were less fixed/more fluid than the original approach assumed. For example, an instructor might adapt training goals themselves during a scenario (training goals within a scenario are not always fixed), change range constraints, decide whether some narrative-breaking, training-specific repair was worth the cost to the training experience in order to maintain it, and so forth.

Table 3 summarizes the new requirements, which are presented in roughly the order of priority that users identified in discussions and assessments with them. Rather than having the system make decisions to keep the scenario within bounds defined by the instructor, the basic functional requirement was to enable the system to adapt its presentation of the scenario to the instructor's goals and intents as the scenario was executing. Requirement #1 reflects this goal, which largely subsumes all of the functional requirements identified earlier in the effort (Table 1). The key insight was that adaptation based on the narrative, training goals, etc. is still desirable, but that adaptation should serve the instructor's intent, rather than static, pre-defined goals and constraints (such as the ability to meet training goals).

Requirement #2 specifies *how* scenario adaptation should be implemented to support users. Because the desired presentation quality can be fluid, there was shift from autonomous control to supervisory control, which allows simulations operators to

Table 3. Revised requirements for scenario adaptation (functional and usage requirements)

Requirement		Description
1	Improve scenario (presentation) quality	Scenario quality is a measure of how closely a scenario that trainees experience reflects the intent of the scenario designer/instructor. A solution must improve quality of presentations (training opportunities) within scenarios, given the instructor's assessment of what is needed to improve training experiences
2	Directable and controllable interventions	Supervisory control, rather than autonomous adaptation, is required. Simulation operators will anticipate undesirable deviations, interpret what should be done, and then activate control programs to intervene if/when undesirable deviations occur
3	Minimize workflow disruption	The simulation-training environment is already being used operationally with existing usage patterns, known technical skill requirements, etc. A solution should, to the extent other goals are met, minimize additional skill or operator-training requirements and not significantly change how models are produced and used today
4	Utilize existing simulated role-players	In addition to minimizing workflow disruptions, scenario adaptation should also employ, to the extent possible, existing simulated role-players, leveraging prior development investment and validation/acceptance of the models that implement the role-players

support more direct realization of instructor intent more easily than low-level control and more flexibly than pre-defined scenario-performance goals.

Requirements #3 and #4 further constrained the technical design space for a solution. Given users' familiarity with the existing simulation, the way it communicated the actions and status of simulated role-players, and the relative maturity of the implementations of those role-players (including validation of some behavior models), any technical solution needed to adopt, to the extent possible, familiar tools, workflows, and models.

Figure 3 illustrates the revised technical design for the scenario adaptation capability, the *Scenario Director*. Because preserving the existing models of role-players was important (Requirement #4), we preserved the Director Architecture design pattern (Fig. 1) but wholly re-implemented the scenario adaptation capability.

To satisfy requirements #1 and #2, we enabled users to specify small control programs (*directives*) that could be applied during a scenario to modulate or over-ride a simulated role-players' native program or to initiate events in the simulation. For example, an operator could specify that when a trainee reached some physical location in a scenario, opposing aircraft should be launched against it with specific instantiation parameters (location, initial heading, speed, altitude, etc.) based on the current situation (the trainee's position and some prior actions). These aircraft might then be modulated by other directives that dictated separation constraints from one another, altitude restrictions, and tactical maneuvers that the instructor desired to see presented to an individual trainee in this scenario.

The simulation operator can specify which directives should be available in a scenario via the Scenario Director user interface. When creating or configuring a scenario, the user can choose from a library of possible directives (directive templates), parameterize them for the scenario (e.g., create a specific altitude restriction for *eagle1* using the altitude restriction template), and then instantiate the directive, which makes

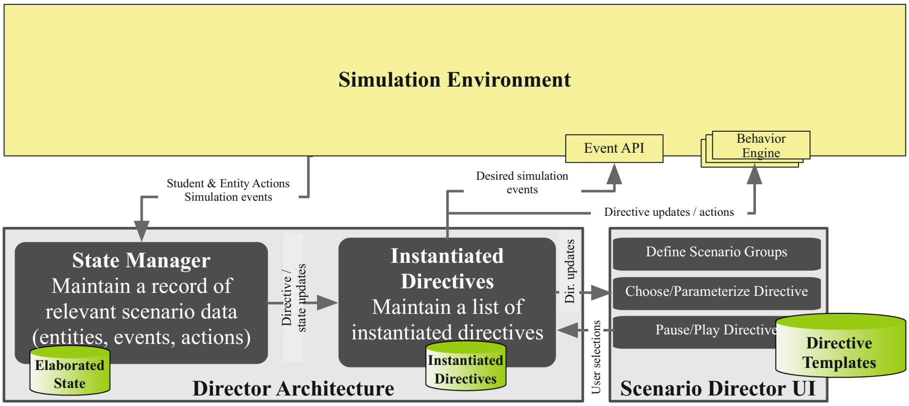

Fig. 3. Scenario Director (revised) software design.

it active in the scenario. Users can define groups (e.g., all aircraft in some region; all aircraft of a particular type or side, etc.). Then, when instantiating a directive, it can be instantiated for the every member of the group, which simplifies the application of directives for a scenario with many simulated role-players (e.g., all blue aircraft in the AO should use this altitude restriction). During run-time, to facilitate meeting instructor intent, individual (and group) directives can be paused or unpaused with a single operator action, enabling an operator to change the course of a scenario in a significant way with a few simple actions in the interface.

Users' are familiar with the technology used to implement simulated role-players in the operational simulation (Req. #3). We took advantage of this familiarity and adopted the behavior modeling language used for simulated role-players for the language of directives. This choice ensures that instructors and operators can specify exactly what interventions they wish to be available. For example, using the same authoring tool that is used for creating simulated role-players, they can also create a directive that encodes exactly what kind of intervention they prefer and the conditions for initiating it. Users add their new directives to the directive template library where other users can then adopt (and modify/customize) those directives.

Compared to the more dynamic and plan-based approach of the Dynamic Tailoring System, this approach to depending on pre-defined interventions is limiting. It requires a lower level of specification from the Scenario Director than the goal-based directability of Dynamic Tailoring, requiring the user to specify not just that an intervention is needed but also exactly what that intervention should be. Because the Scenario Director is limited to making changes to only exposed parameters, it range of direction is also limited. For example, if an aircraft's partner is not specified in one of the parameters in the scenario, then the Scenario Director cannot make dynamic changes to that pairing.

Overall, however, this approach has been advantageous. Because direct operator control of simulated role-players is already expected, most important parameters for control are already exposed, providing sufficient levers for direction. The challenge of codified interventions is offset by understandability and predictability. Because the Scenario Director adapts behavior using the same language and mechanisms that human operators use, operators both understand what the Scenario Director is doing and have an existing model for predicting what will happen when direction is initiated, including how to recover from surprises or errors if an intervention takes a scenario off-course non-productively.

The long-term impacts on performance have yet to be measured and initial assessments of trustworthiness and transparency are sometimes not correlated with long-term performance [18]. However, the user community has taken up the Scenario Director implementation of scenario adaptation. It has now met all milestones for transition to operational simulation. The software has been delivered to the simulation development team and is now incorporated in its development releases. This Scenario Director capability will be further refined and extended by the simulation-development team for a major version to be deployed in 2018.

6 Conclusion

This paper has presented a specific use case of the use of intelligent automation to support scenario adaptation for simulation-based training. We initially focused on functional requirements automatic run-time adaptation of scenarios, which was both consistent with prior requirements in other training domains and appeared apt for the fast-moving, high-workload tactical-air training environment. After initial prototyping and demonstrations to potential users, we identified additional constraints on solutions and usage requirements. These additional requirements led to a near-complete redesign and reimplementation of the scenario adaptation capability.

The new capability traded relatively simple and fast pre-scenario configuration and run-time autonomy for both greater pre-scenario configuration (including, possibly, the creation of a new control program/directive) and a more supervisory, rather than autonomous solution for run-time adaptation. Although users lost some capability in this change, they felt these losses were offset by the benefits of greater transparency (understanding what the system would do and when) and more flexible. Additionally, because the current environment of training delivery already includes personnel who can direct and control simulated role-players, the emphasis in the new solution was to attempt to make these staff more effective in delivering the training experience that the instructor desired, reducing workload and enabling more consistent training delivery. We expect that in training environments where staff is available and specific performance measures are not defined for individual training scenarios, an extensible, supervisory control approach to scenario adaptation is likely to be preferable to a more autonomous one that is standardized on the curriculum.

Acknowledgments. This work is supported by the Office of Naval Research project N00014-1-C-0170 Tactical Semi-Automated Forces for Live, Virtual, and Constructive Training (TACSAF). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Defense or Office of Naval Research. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. We would like to thank collaborators and sponsors at NAWCTSD and ONR who have provided insights and operational perspectives in the development of the capabilities discussed herein: CDR Brent Olde, Ami Bolton, Melissa Walwanis, and Heather Priest.

References

1. Wray, R.E., Woods, A.: A cognitive systems approach to tailoring learner practice. In: Laird, J., Klenk, M. (eds.) Proceedings of the Second Advances in Cognitive Systems Conference, Baltimore, MD (2013)
2. Parasuraman, R., Riley, V.: Humans and automation: use, misuse, disuse, abuse. Hum. Factors **39**, 230–253 (1997)
3. Parasuraman, R., Wickens, C.D.: Humans: still vital after all these years of automation. Hum. Factors **50**, 511–520 (2008)

4. Sheridan, T.B.: Humans and Automation: System Design and Research Issues. Wiley, Hoboken (2002)
5. Parasuraman, R., Sheridan, T.B., Wickens, C.D.: A model of types and levels of human interaction with automation. IEEE Trans. Syst. Man, Cybern. - Part A Syst. Hum. **30**, 286–297 (2000)
6. Parasuraman, R.: Designing automation for human use: empirical studies and quantitative models. Ergonomics **43**, 931–951 (2000)
7. Lee, J.D., See, K.A.: Trust in automation: designing for appropriate reliance. Hum. Factors: J. Hum. Factors Ergon. Soc. **46**, 50–80 (2004)
8. Hoff, K.A., Bashir, M.: Trust in automation. Hum. Factors **57**, 407–434 (2014)
9. Wray, R.E., Priest, H., Walwanis, M.A., Kaste, K.: Requirements for future SAFs: beyond tactical realism. In: 2015 Interservice/Industry Training, Simulation, and Education Conference. NTSA (2015)
10. Wray, R.E., Folsom-Kovarik, J.T., Woods, A., Jones, R.M.: Motivating narrative representation for training cross-cultural interaction. In: Proceedings of the 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015. Springer, Las Vegas (2015)
11. Vygotsky, L.S.: Mind and Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge (1978)
12. Riedl, M.O., Stern, A., Dinini, R., Alderman, J.: Dynamic experience management in virtual worlds for entertainment, education, and training. Int. Trans. Syst. Sci. Appl. Spec. Issue Agent Based Syst. Hum. Learn. **4**, 23–42 (2008)
13. Magerko, B.: Evaluating preemptive story direction in the interactive drama architecture. J. Game Dev. **3**, 25–52 (2007)
14. Wray, R.E., Lane, H.C., Stensrud, B., Core, M., Hamel, L., Forbell, E.: Pedagogical experience manipulation for cultural learning. In: Workshop on Culturally-Aware Tutoring Systems at the AI in Education Conference Brighton, England (2009)
15. Folsom-Kovarik, J.T., Newton, C., Haley, J., Wray, R.E.: Modeling proficiency in a tailored, situated training environment. In: Behavior Representation in Modeling and Simulation (BRIMS) Conference, Washington, DC (2014)
16. Wray, R.E., Woods, A., Haley, J., Folsom-Kovarik, J.T.: Evaluating instructor configurability for adaptive training. In: Proceedings of the 7th International Conference on Applied Human Factors and Ergonomics (AHFE 2016) and the Affiliated Conferences. Springer, Orlando (2016)
17. Dillon, D.: Beyond usability: process, outcome and affect in human-computer interactions. Can. J. Libr. Inf. Sci. **26**, 57–69 (2002)
18. Schaefer, K.E., Chen, J.Y.C., Szalma, J.L., Hancock, P.A.: A meta-analysis of factors influencing the development of trust in automation: implications for understanding autonomy in future systems. Hum. Factors: J. Hum. Factors Ergon. Soc. **58**, 377–400 (2016)