A Cognitive Systems Approach to Tailoring Learner Practice

Robert E. Wray WRAY@SOARTECH.COM
Angela Woods ANGELA.WOODS@SOARTECH.COM
Soar Technology, Inc., 3600 Green Court Ste 600, Ann Arbor, MI 48105 USA

Abstract

With the emergence of low-cost, realistic simulation technologies, learner practice of complex cognitive skills (e.g., aircraft piloting, tactical observation, and cross-cultural conversation) is feasible. Tailored, guided practice has been shown to be more effective than practice alone but practice mediated by human experts is expensive and not scalable. As a consequence, there exists significant interest in developing algorithms and technologies for autonomously guiding and adapting learner practice. We describe an integrated cognitive systems approach to developing an adaptive training technology for tailored practice. We build on the pivotal foundations cognitive architectures have provided historically for adaptive instructional technologies (especially Intelligent Tutoring Systems). We also identify new requirements that, we contend, intensify the need for a cognitive systems approach. We describe how a specific cognitive systems approach is contributing to the satisfaction of these new adaptive-training requirements and the creation of flexible, reusable adaptive training technology.

1. Introduction

Simulation-based training is becoming a cornerstone of adult skill training in both the commercial and military domains. For example, simulation technology is used throughout all the branches of the US military, at almost all echelons, and for a broad range of military roles and missions (Fletcher, 2009). Well-designed simulation enables more frequent and sustained learner practice. Learners practice in a fail-safe environment. The costs of repeated training experiences can be much lower, due to much lower comparative cost of simulation-based training than the real world (Fletcher & Chatham, 2010). Additionally, simulated entities can provide a realistic environment of interactions without incurring the substantial costs of human role players to support training.

Although simulation is powerful tool for enabling practice, effective instruction requires more than providing a realistic practice environment. The most effective learning is hypothesized to occur when a practice situation is tailored to a learner’s individual ability at that point in time (Vygotsky, 1978) and guidance and feedback are provided as practice progresses (Clark, 2009). Although the specific, empirical learning gains from human one-on-one instruction are not as great as once thought (Van Lehn, 2011), there is ample evidence that well-trained instructors working with a student or small group of students can deliver highly effective training outcomes (Clark, 2009).

Delivering highly individualized, effective instruction and practice via human tutors and mentors, however, is generally not scalable or cost-effective. There is presently significant interest in research and development of software systems that can approximate the impacts of human instruction in computer-aided learning environments including simulation-based practice environments. For example, in the United States, there are initiatives sponsored by the Department of Defense and National Science Foundation in educational technology for Science Engineering Mathematics and Technology (STEM) learning. Further, the National Academies of Engineering has identified "personalized learning" as a grand challenge for 21st century engineering (National Academy of Engineering, 2008).

The goal of researching and developing software systems that approximate human instruction is not new to the cognitive systems community. Many Intelligent Tutoring Systems have been built on the foundation of cognitive architectures, such as the Andes Physics Tutors (Van Lehn et al., 2005) and ACT-R-based cognitive tutors (Anderson et al., 1995). These systems use the integrated cognitive capabilities of underlying cognitive architectures to meet requirements for the learning systems. Examples include distinctions between declarative memory, where facts like the sums of small numbers or important constants (e.g., force of gravity) are stored and accessed and the procedural rules of the domain (e.g., commutative property, formulas expressing physical laws). Using these kinds of representations enable the system to model (or at least reproduce) expert solutions to problems and to attempt to diagnosis specific student errors (Corbett & Anderson, 1995).

Today, emerging requirements for learning systems are being identified that intensify the need for integrated cognitive systems approach. These requirements include:

1. The need to support learning in domains that have unpredictability and ambiguity, and lacking well-defined rules for interpreting a domain state or taking action;
2. Highly dynamic domains, that require nearly continuous monitoring and, potentially, adjustment as the learner progresses through a learning exercise;
3. Instructional flexibility, enabling a system to respond to different students (or the same student in a similar situation at a different time) with different instructional approaches and tailoring actions; and
4. Multi-dimensional interpretation of an individual learner's affective and cognitive state, enabling diagnosis beyond right and wrong to include motivation and engagement in assessing what instructional actions or interventions are necessary.

We are currently researching and developing a general software system designed to address these new requirements while also providing a reusable software framework that can be used in many different domains. The Dynamic Tailoring System (DTS) supports tailoring of practice in simulation. It is implemented using the Soar architecture (Laird, 2012). The DTS uses Soar as an agent architecture (Wray & Jones, 2005) rather than to model an instructor explicitly. However, as detailed further below, the integrated cognitive foundation of Soar has contributed significantly to the approaches and solutions we are taking to meeting the new requirements introduced above.

Below, we first describe the overall structure of the DTS and introduce recent or on-going applications in three different domains: intercultural conversation, tactical observation, and tactical aircraft piloting. We then discuss each of the new requirements, describing their importance, implications for system capabilities, and how an integrated cognitive systems approach, as exemplified by the DTS, is being used to realize or explore the requirement, with examples drawn from the application domains.

2. Overview of the Dynamic Tailoring System

Dynamic tailoring falls into the broader category of *experience manipulation*, an adjustment of the normal behavior of simulation elements to achieve an objective. *Pedagogical experience manipulation* concerns how one can intrinsically adjust a learning environment and simulation to promote learning and facilitate pedagogical goals (Lane & Johnson, 2008). The implementation of the Dynamic Tailoring System presumes that the system's pedagogical experience manipulation will usually be used in tandem with extrinsic (outside the simulation) supports, such as direct instruction and intelligent tutoring. The DTS also works in conjunction with distinct processes that that select and instantiate problems and scenarios (Magerko, Stensrud, & Holt, 2006). Thus, the Dynamic Tailoring System focuses on "inner loop" (Van Lehn, 2006) tailoring and a separate "outer loop" process chooses and instantiates particular practice exercises.

Because the space of possible interventions from tailoring is large (and growing, as basic researchers explore and evaluate new methods for supporting learning), the design of the Dynamic Tailoring System is built around a framework that organizes possible interventions and adaptations (Wray et al., 2009). Table 1 summarizes primary framework elements.

Table 1: A Summary of Dynamic Tailoring Goals and Mechanisms.

Goals of Tailoring	**Scaffolding** adapts content to support practice directly (Davis & Miyake, 2004). Paired with **fading** or the gradual removal of scaffolds that are no longer necessary as learners move to higher levels of ability (Pea, 2004).	**Engaging**: Learner errors may derive not only from a lack of knowledge but arise from affective states related to attention, stress, and motivation. Engagement adaptations attempt to increase engagement or to reduce stress.
	Challenging is the inverse of scaffolding, introducing content adaptations that make fine-grained components of practice more difficult.	**Individualizing**: Practice environments need to adapt how the simulation works to support the diversity in preferences, technical competencies, and motivation (Heeter & Winn, 2008)
Mechanisms of tailoring	**Outcome manipulation**: The effects of simulation actions can be modified or modulated by the tailoring system. E.g., scaffolding outcome manipulation might amplify negative effects of errors likelihood of learner detection.	**Choice manipulation is m**anipulating the options and actions available to a learner. Examples include direct modifications of the actions available to the leaner as well as adaptations to the sequence and relationships of events (Magerko, 2007; Riedl et al., 2005).
	Character utterance or gesture: Character actions can be tailored in response or anticipation of learner. Character behaviors are usually implemented distinctly from the events in the simulation and will likely differ in implementation from outcome manipulations.	**Simulation/Event manipulation**: Any pre-emptive event introduced to support pedagogy. E.g., varying signal-to-noise ratio of learning-relevant cues, introducing environment events that reinforced pedagogical feedback or bring the learner's attention to the environment.
	Gameplay manipulation: Tailoring may include manipulations that change the way a simulation is experienced. E.g., adding or removing an explicit representation of progress and status, enabling access to external (non-game) content to amplify or aid the pursuit of learning objectives, and changing the user interface to accommodate preferences and technical literacy of individual learners.	

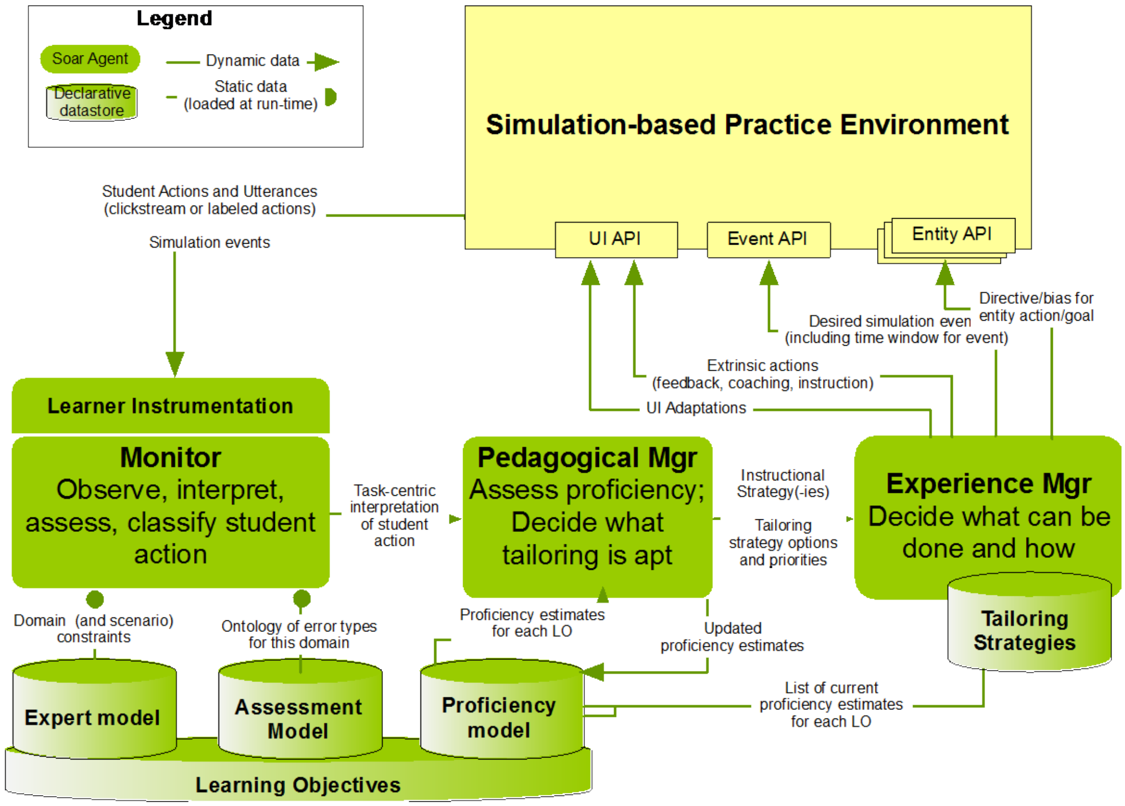

Figure 1. Software Architecture of the Dynamic Tailoring System.

The framework enumerates general reasons to use tailoring strategies and the classes of tailoring strategies that apply to simulation-based learning environments and thus defines a space of functional requirements that a dynamic tailoring software system should address. The intent of the framework is to organize exploration of tailoring methods and generalizations about learning outcomes without tying experiment results to specific interventions in a particular system.

2.1 Cognitive-Architecture-based Dynamic Assessment and Tailoring

Figure 1 illustrates the systems architecture of the Dynamic Tailoring System. Similar to an ITS, the system tracks learner actions and maintains a student model. An *expert model* describes correct behavior, an *assessment model* facilitates error recognition, categorization and mapping to active learning objectives, and a *proficiency model* enables long-term tracking of performance against learning objectives. Three distinct agents accomplish dynamic tailoring: Monitor, Pedagogical Manager, and Experience Manager.

The basic flow of control starts with the expert model that defines appropriate and inappropriate actions. The Monitor draws from the expert model and other available measures of the learner (e.g., physiological measures relating to boredom/stress) to assess learner actions. Based on the assessment, the Pedagogical Manager updates the estimated proficiency of the student and sets preferences for tailoring, such as desired levels of difficulty or helpfulness for individual learning objectives. The Experience Manager then evaluates the preferences in terms of available tailoring options and determines a course(s) of action.

The Monitor and Experience Manager are implemented via Soar and, as we outline further next, employ native mechanisms of Soar to achieve the desired functionality. The Pedagogical Manager is currently implemented in Java, but we are re-implementing the Pedagogical Manager in Soar to accommodate an emerging need for context-mediated decision making in the Pedagogical Manager (Section 3.2.3). We next outline the role cognitive architecture and other algorithms play in the existing implementation of this software system. Section 3 will then describe details of some of the approaches and implementations to illustrate the roles of integrated cognitive components in realizing solutions to key requirements.

2.1.1 Monitor

The Monitor's responsibility is to observe learner actions (and, when available, affective state), interpret those actions and states in the context of the learning situation, and then to assess the learner's behavior in terms of active learning objectives and relevant indices (e.g., estimated level of arousal). Interpretation processes are not specific to the domain. Instead, the Dynamic Tailoring System supports a declarative expert model representation loosely derived from constraint-based tutoring (Mitrovic & Ohlsson, 1999). Errors are detected in the Monitor by identifying conflicts between the learner's actions and constraints specified in the model. Constraint-based models are easily authored; an expert model representation can be created quickly without needing detailed knowledge of the Monitor implementation that performs the constraint checking.

The monitor classifies errors (errors of omission, commission, etc.) by comparing its interpretation of learner behavior with an Assessment Model that defines the class of an error (e.g., skipping a required step results in an "omissions error" label). The Monitor takes advantage of Soar's efficient pattern matching capability to match constraints and identify constraint violations. It uses Soar's problem space decomposition function to organize and to switch between scenario and learner performance "contexts" specified in the expert model.

2.1.2 Pedagogical Manager

The Pedagogical Manager is the newest DTS component and is presently implemented in Java. The Pedagogical Manager's primary roles are to maintain the model of learner proficiency and to specify high-level preferences for tailoring.

The Dynamic Tailoring System's proficiency model is derived from the classic work on SHERLOCK (Katz et al., 1998). Learner proficiency is modeled via a "fuzzy variable" with five discrete but overlapping states, which map loosely to progressive learning/mastery of the concept/skills. The Dynamic Tailoring System's proficiency model extends the original SHERLOCK approach to enable multiple levels of learning objectives, different responses to individual learning objectives deriving from one action assessment, and an algorithm for updating variables that is sensitive to the learning context (see Section 3.2.3).

After each learner action and assessment, the Pedagogical Manager updates the fuzzy variable. The fuzzy variable is represented by a vector: <*none, limited, partial, non-automated, fully-developed*> where the values of the indices always total 100. The semantics of each value in the vector corresponds nominally to the likelihood of the learner's proficiency for the learning objective corresponding to that index. This approach offers a simple, well-defined operational semantics of the fuzzy variable (i.e., instead of a theoretical commitment to some externally measured or validated level of proficiency). As the learner acts and the Monitor assesses those acts, the Pedagogical Manager then updates the estimates for each learning objective node in the proficiency model. Because individual proficiency vectors are tracked for each learning objective, tailoring responses can target these fine-grained estimates, as well as an aggregate estimate of proficiency.

Rather than mapping proficiency directly to tailoring actions, the Pedagogical Manager instead specifies "tailoring preferences" for each learning objective. The current implementation supports three kinds of preferences (HELPFULNESS, SIMPLICITY, and PREDICTABILITY), which the Pedagogical Manager specifies for each learning objective and can change at any time. For instance, for a student who is demonstrating increasing competence with respect to some learning objective, the Pedagogical Manager may choose to reduce HELPFULNESS and SIMPLICITY for that learning objective, supporting a rough approximation of cognitive apprenticeship (Collins, Brown, & Newman, 1989). However, as we discuss further in Section 3.2.3, the Pedagogical Manager's design anticipates multiple instructional approaches and the ability to choose and dynamically switch the instructional approach.

One motivation for moving to a Soar implementation of the Pedagogical Manager is that the complexity of the decisions that the Pedagogical Manager needs to make is growing more complex in two ways: 1) mediating between extrinsic stimuli (such as explicit instruction or warnings provided by an ITS) and the intrinsic adaptations of tailoring and 2) mediating between affective and domain tailoring actions. For example, when a novice learner makes a mistake, it may be more beneficial to the learner to receive direct feedback from an ITS or "Coach" (Lane & Johnson, 2008) than to receive intrinsic feedback. However, for a more experienced learner, receiving cues from "within the environment" may be more appropriate. The Pedagogical Manager is the component that makes such decisions.

2.1.3 Experience Manager

The Experience Manager receives tailoring preferences from the Pedagogical Manager and chooses an action or plan based on the learning context. For example, a tailoring action that provides few contextual cues (HELPFULNESS=LOW) and introduces uncertainty in feedback (PREDICTABILITY=LOW) can be used to challenge a proficient learner.

The Experience Manager is implemented in Soar and uses Soar as a plan execution system, roughly following Magerko's (2007) Interactive Drama Architecture design, which tailors user experience for games and learning. Tailoring actions are represented by preconditions that specify applicability. Using Soar's decision process, available tailoring actions (what options can be brought to bear) can be sorted according to current tailoring preferences using Soar's built-in preference representation. The choice (or collection of a series of choices) is initiated via communication with the simulation. Although currently the implementation is limited to choosing pre-defined tailoring actions/plans, longer-term, we plan to extend the Experience Manager to construct new tailoring actions (via, e.g., planning as in Thomas & Young, 2011).

2.2 Example Applications

Figure 2. Modulating Character Affect for Pedagogy.

The implementation of the Dynamic Tailoring System supports reuse of the core components across multiple applications. As we discuss further in the next section, one advantage of working within a cognitive architecture to implement tailoring capability is that we can draw on representations and processes at the fixed, architectural level to guide the implementation of specific decision making processes in the individual components. We then expect that the architectural foundation will then enable efficient reuse across different practice domains. To highlight this aspect of the implementation, we briefly introduce three applications of the Dynamic Tailoring System.

2.2.1 Intercultural Conversation Skills

An early application of the Dynamic Tailoring System is within an intercultural skills training system called the Cultural Meeting Trainer (CMT). The CMT is a lightweight 2D graphical user interface to the Institute for Creative Technologies' bi-lateral negotiation (BiLAT) training system (Kim et al., 2009). The learner engages in a simulation of social and business interactions with virtual Iraqi counterparts, enabling practice of intercultural communication.

The CMT focuses on the cultural aspects of "small talk" and trust building during social periods of meetings. The goal of each interaction is to "chat" with an individual character and gain his/her trust sufficiently to move into a formal negotiation. Five tailoring strategies were designed and implemented for this domain, using the Dynamic Tailoring System to organize the different methods and to deliver them as appropriate during individual practice sessions. For example, as shown in Figure 2, via a "character affect tailoring" strategy, the DTS modulates the appearance of emotional responses of the character depending on pedagogical context.

2.2.2 Observational Skills

A second application of the DTS is within a training application in which US Marines observe a village from a Virtual Observation Platform (VOP). The VOP is inspired and informed by successful "live" training programs (e.g., Schatz et al., 2010) in which Marines learn to construct a general "baseline" of understanding from sustained attention to activities in a "village" (populated by human role players). Needed skills range from low-level signals (recognizing the proxemics and kinesics of individual villagers), to recognizing and categorizing quotidian and unusual events, to developing an abstract mental representation of the "patterns of life" (Schatz et al., 2012) within the village.

The VOP itself is being developed by DSCI MESH and includes sensors that detect gaze and the visual attention of learners, as well as an immersive simulation environment (Schatz, et al., 2012). In this domain, we are using not only the core DTS software implementation, but also several of the previously developed tailoring strategies from the CMT. New tailoring strategies are also being applied, such as the injection of narrative events and systematic variation of signal-to-noise in perceptual cues (Schatz, et al., 2012). A major difference in this domain, versus the conversational domain, is that a learner is much less visibly "active" than in other domains, needing to observe and interpret over several minutes. As a consequence, we are supplementing the Dynamic Tailoring System with Learner Instrumentation Middleware designed to support system understanding of attention and other latent factors (Wray, Folsom-Kovarik, & Woods, 2013). We will return to this design requirement in Section 3.2.4.

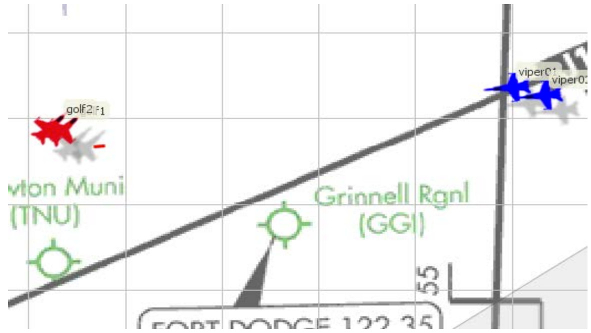

Figure 3. Adapting opponent tactics in response to learner errors.

2.2.3 Tactical Piloting Skills

The most recent application of the DTS is for training of aircraft pilots in simulation. The application focuses on the training of platform capabilities and tactics for US Navy aviators, after they have mastered basic flight skills. Some of this training takes place in advanced, immersive flight simulators, integrated with distributed simulation engines that model aircraft flight dynamics, sensors and weapons, and the behavior of simulated entities (Semi-Autonomous Forces, or SAFs) in a scenario. The goal of dynamic tailoring in this environment is to provide a realistic and pedagogically appropriate training experience while ensuring that mandated training situations are presented during the training, minimizing the need for human intervention during the scenario.

As an example, in the situation illustrated in Figure 3, the learner (viper01) aircraft missed a mission-critical coordination event that leaves viper01 and viper02 vulnerable to attack from the red aircraft (golf1 and golf2). If the red aircraft successfully attack the learner (as they have initiated in this illustration), the learner will miss a required training event designed to occur in this scenario subsequent to this initial engagement. In response, the DTS dynamically modifies the mission of the red SAFs, directing them to attempt to fly thru the learner's defense, rather than to engage.

This domain has required novel tailoring strategies, although the primary DTS components were applied with relatively minor changes. An exception was the tailoring systems' response time. This domain requires much faster responses than the previous two example domains, which led to some performance optimizations for highly dynamic domains (Section 3.2.2).

3. Cognitive System Requirements for Adaptive Learning Systems

Having introduced the functional requirements for dynamic tailoring and a summary of an implementation of dynamic tailoring within a cognitive architecture, this section discusses the advantages and rationales for taking a cognitive-systems approach to the dynamic tailoring problem. We first summarize some of the characteristics of dynamic tailoring that make it representative of the class of problems for which integrated cognitive systems are targeted. We then outline additional functional requirements for dynamic tailoring which further highlight the role of cognitive systems approaches in extending and refining an implementation of dynamic tailoring that is general enough to be applied to many different classes of learning domains.

Table 2: Dynamic tailoring environments and tasks manifest cognitive architecture requirements.

Characteristic	**Dynamic Tailoring Implications**
C1 Complex Environment	The simulation-based environment is complex, requiring understanding and interpretation of simulation events, learner actions, and estimation of latent states. The complexity of the environment is managed via large stores of symbolic knowledge (R1, R3), multiple levels of abstraction and generality (R4), rich action representations (R7) and multiple modes of deliberation and reflection (R8, R9).
C2 Dynamic Environment	Both the simulation and the learner act independently of the tailoring system. Modularity in the knowledge representation (R2), multiple levels of abstraction (R4), rich action representations (R7), and multiple modes of deliberation (R9) help the DTS be responsive to a dynamic domain.
C3 Task-relevant Regularities	Learner activity is often episodic (scenarios) and, by instructional design, learner tasks recur resulting in similar situations for interpretation and tailoring. Modular, multi-layered, symbolic knowledge (R1,R3, R4, R5) enables codification, recognition, and response to these regularities. Although not implemented today, long-term, as discussed further below, we expect the DTS will also have the capability to learn to recognize regularities autonomously.
C4 Other Agents	Other agents include not only individual learners, but also instructors (who may sometimes be present). One of the advantages of general, symbolic representations (R1, R3) is that these representations can be presented readily to an instructor, to support transparency and rationales for tailoring actions.
C5 Complex, Diverse, Novel Tasks	Task diversity is relatively narrow but includes interpretation and assessment and selection (and composition) of tailoring actions (R3, R5, R7). The DTS is designed with encapsulated domain-specific and domain-general knowledge components, so that diversity of applications (different learning domains) can be supported via a common/reusable software environment.
C6 Complex and Limited Interactions	The environment is only partially observable. Although software engineering can make the simulation state transparent, key latent variables include proficiency estimates and the learner's affective state. This complexity is primarily addressed via large, modular stores of symbolic knowledge (R1,R3,R4,R5) although we are also exploring the use of modality-specific representations (R2) for assessing affective state and estimating proficiency.
C7 Limited Agent Resources	Because the learner is taking action in the environment with the tailoring system, the system must respond very quickly, requiring it to sometimes trade diagnostic specificity (exactly why the student made an error) for a more shallow interpretation (e.g., the class of error). The available spectrum of deliberation (R9) within the cognitive architecture in particular supports evaluation of these trade offs.
C8 Long-term Existence	This criterion is less critical for dynamic tailoring. However, dynamic tailoring must follow a learner across multiple practice experiences and changing learner capability. The changing abilities of the learner require meta-cognitive knowledge (R8), in addition to other symbolic knowledge, to reflect on learner state and system action in response to that state.

3.1 Cognitive System Requirements for Adaptive Learning Systems

As outlined above, we are using Soar as a capable functional substrate on which to build these capabilities but we are not attempting to use Soar to model the way in which a human tutor would recognize and classify student actions or deliver feedback. However, we maintain that the basic challenge of dynamic tailoring is consistent with the environment, task and task characteristics that motivate cognitive architectures.

To illustrate, Laird and Wray (2010) summarize eight environment and task characteristics that lead to the derivation of eleven cognitive-architecture requirements. They then map individual requirements to specific environment and task characteristics. Table 2 lists the environment and task characteristics from the analysis and outlines how these characteristics are manifest in computational approaches to dynamic tailoring and some of the cognitive architectural requirements that are important in realizing a solution (the R labels represent the requirements as identified in their analysis). Although the current implementation does not learn from observation or experience (requirements R10 and R11), all the non-learning cognitive requirements outlined in the analysis are reflected in the task characteristics imposed by dynamic tailoring.

Because the individual components of the DTS were developed somewhat independently of one another and have different functional and knowledge representation requirements, it was convenient, from a software engineering perspective, to encapsulate these functions into distinct Soar instances. It is an open question whether one instance of the architecture would be preferable to three distinct instances for Monitor, Experience Manager, and Pedagogical Manager.

3.2 Emerging Requirements for Dynamic Tailoring

As we investigate the ways in which dynamic tailoring can be used to support learners, we have experienced the need to support and extend the basic DTS software architecture to meet additional requirements. This section details some of these "new requirements" and the ways in which cognitive architecture has provided or is providing insights and capability for meeting the requirements. We begin with two requirements not explicitly considered in the early design of the DTS, which have largely been addressed in the implementation presented above. We then discuss a third requirement that is being actively developed and another emerging requirement, which we are exploring via on-going design and prototyping. The intent is to survey the way in which some software functional requirements, which may not be evident when a system is first designed, can be more readily and conservatively met using a cognitive systems approach.

3.2.1 Supporting Learning in "Ill-Defined Domains"

Many of the most successful and well-known Intelligent Tutoring Systems were applied in domains such as math and physics, where problems can be clearly and unambiguously stated and there exist well-defined procedures for solving the problems. The first application of the DTS was in a medical triage domain that largely fit these criteria (Magerko, Stensrud, & Holt, 2006). However, many interesting learning domains include unpredictability and ambiguity, partial observability, and may often lack well-defined rules for interpreting a domain state or taking action. Applying traditional ITS approaches to these so-called "ill-defined domains" has proven challenging (Fournier-Viger, Nkambou, & Mephu, 2010; Ogan, Wylie, & Walker, 2006).

The approach we have taken to expert models reflects the challenges of interpretation of behavior in an ill-defined domain. Constraints in the expert model form an implicit "envelope" that bounds behavior. As long as learner behavior remains within the envelope, the system allows a wide-range of learner actions. For example, in the cultural meeting domain, many different student utterances are allowed at each phase of the meeting. Some of these utterances can be associated with specific phases of the meeting; for example, formal greetings and signs of respect typically occur in the introductory phase of the meeting. The expert model expresses constraints that certain kinds of utterances should occur in this phase of the meeting, but does not directly forbid (mark as an error) the use of introductory utterances at later phases of the meeting. If the learner failed to remove the avatar's sunglasses on entering the meeting (a sign of respect), then removing them after formal introductions have concluded is still likely acceptable. This approach allows the DTS to track student progress and note errors, but not force the learner into a rigidly enforced sequence of steps. This approach also potentially mitigates the frustration and loss of engagement learners experience when a learning system scaffolds based on improper diagnoses of an error (Puntambekar & Hübscher, 2005).

The limitation of this approach, of course, is that the learner may need more guidance and the system may need better diagnostic power in order to choose the most apt tailoring option(s). Working around this limitation is where the advantages of using the cognitive systems approach became more apparent. The Monitor's constraint checking could be performed by purpose-built algorithms or simpler pattern-matchers. However, because the Monitor is implemented in Soar, we can readily extend its overall capability by immediately taking advantage of additional capabilities available within the architecture.

For example, the modeling language features design patterns for switching between learning contexts (introductory phase of a meeting vs. small-talk phase). We designed these patterns to be simple and authorable by instructors, who typically lack programming skills; thus, the system trades model expressivity for authorability. In other applications, such as the tactical air domain, the limited expressivity requires enumeration of many individual contexts. For this application, we are exploring richer knowledge of changing learner context in a more general way than the fixed rules in the language. Because the constraints and context-switching knowledge is already embedded in the Monitor, this research can bring the full power of the architecture to bear on the problem. We take a similar approach, supplementing the knowledge from constraint errors and satisfactions using the expert model with additional knowledge to provide dynamic guidance during learning (Wray, Woods, & Priest, 2012).

Longer-term, we expect that some combination of "surface evaluation" and fine-grained assessment of the learner's process will be desirable (Fournier-Viger, Nkambou, & Mephu, 2010). Finer-grained diagnosis could be achieved within the DTS using model-tracing methods (e.g., Anderson, Corbett, Koedinger, & Pelletier, 1995), which have often been developed within cognitive architectures. Similarly, the existing coarse-grained error diagnosis of the current DTS could be coupled with domain-specific knowledge that responded directly to model-defined procedural error patterns (e.g., Brown & Van Lehn, 1980) or domain misconceptions (e.g., Vosniadou, Skopeliti, & Gerakaki, 2007) to strengthen the instructional response of the system when more refined diagnosis is available. Thus, rather than requiring a complete process model (which is often prohibitively expensive) or a constraint-based model (which limits diagnosis), using the cognitive architecture supports inclusion and integration of both, depending on application and model requirements. Additionally, in the context of a complete instructional system, direct assessment and reflective practice can be used to improve proficiency estimates (e.g., Wray & Munro, 2012). Thus, using multiple techniques to estimate learner state may reduce the need for a detailed process model of learning state for use in tailoring practice.

3.2.2 Adapting to Learners in Highly Dynamic Domains

Another feature of many practice domains, in comparison to traditional ITS domains, is the highly interactive and dynamic aspect of the domains. Both the observation domain and the tactical air domain have many actors taking action and interacting with the learner in a way that sometimes requires near-immediate responses from the learner. An obvious example in the tactical air domain is the maneuvering of aircraft that intend to shoot one another down ("intercepting").

The high-frequency dynamics of the environment impose additional constraints on dynamic tailoring because the tailoring system may wish to intervene before the learner or the simulation takes an action. In Figure 3, for example, a red aircraft has launched a missile a moment after reaching the effective range of its weapon. Dynamic tailoring must be fast enough to respond to this condition if delaying missile launch is a better option for the learner. The response time required for dynamic tailoring thus reinforces the need for integrated approach. All three stages of the decision making process must be efficient and sufficiently integrated that information can pass quickly and succinctly from one component to another.

As an example of what happens when decision-making is not responsive enough, we recently performed a series of redesigns and performance optimizations for the Monitor. Early implementations of the Monitor placed the contents of the expert model directly into Soar's working memory. This strategy is convenient from a development perspective and not necessarily computationally expensive, as suggested in recent evaluations of Soar performance (Laird, Derbinsky, & Voigt, 2011). However, the Monitor implementation, due to similarity in the structures that were being placed into memory and constraint evaluation approaches that required searching for particular substructures in the constraint set, began to experience performance limitations attributable to expensive matching (Tambe, Newell, & Rosenbloom, 1990). We refactored so that less constraint structure was available in working memory at all times and additional information about a constraint could be retrieved deliberately when needed. This situation was a reminder of Newell's (1990) maxim to "listen to the architecture" when facing design tradeoffs in implementation.

A more recent example of optimization focused on reducing the size of the memory needed by the Monitor. The Monitor marked satisfied constraints but left those structures in memory to give the system access to a learner history. The short-term solution was to move this history out of agent memory, which eliminated issues with unbounded memory size. This solution was practical from an engineering perspective but, from an integrated cognitive systems perspective, inefficient and inelegant, because the history is no longer (readily) available to the Monitor when it may need it. Our longer-term plan, which also reflects "listening to the architecture," will be to record these events via Soar's episodic learning and memory (Laird, 2012). The advantage of this approach would likely be that the storage and retrieval of episodes is already highly optimized within the architecture (Laird & Derbinsky, 2009), requiring no external memory access or specialized storage interfaces to achieve this capability.

3.2.3 Enabling Flexibility in Instructional Approach

There are presently competing theories about the best ways to structure and to organize practice to support learning (e.g., Tobias & Duffy, 2009). Further, active research attempts to identify ways to support the development of learner skill in a domain as well as maintaining affective states that promote or support that learning (Woolf et al., 2009). However, the vast majority of computer-aided learning systems assume a few (often one) instructional approaches in their approach to delivering instruction. Rather than committing the Dynamic Tailoring System to one instructional paradigm versus another, we hypothesize that instructional flexibility is key requirement for adaptive learning systems. That is, the system should be able to respond to different learners (or the same learner in a similar situation at a different time) with different instructional strategies and instructional actions, chosen by the system for that moment.

The Pedagogical Manager is the component in the DTS that makes the decision about what instructional approach is most apt for the current situation. We are currently investigating how to support an approach styled on cognitive apprenticeship, mentioned previously, and another based on "preparation for future learning" (Bransford & Schwartz, 1999).

In the cognitive apprenticeship strategy, learners with low estimated proficiency for a learning objective receive simpler problems with predictable outcomes. The Pedagogical Manager specifies general preferences for tailoring that will explicitly help the learner connect actions and feedback to learning objectives, either through an extrinsic dialog or intrinsic feedback within the environment. For example, in the Virtual Observation Platform domain, a virtual observation team can be helpful to the learner and provide hints and fill in missed details that the learner may have missed. As the learner demonstrates increasing skill, these supports are faded, by preferring less helpful and more intrinsic feedback and by making the learning situation more complex (e.g., more individuals to monitor in the village) and less predictable (more ambiguity in the likely meaning of character behaviors). The "preparation for future learning" strategy offers much less scaffolding initially, allowing the learner to explore the environment without guiding feedback and with much more of the complexity and unpredictability of the real-world environment.

At the implementation level, "listening to the architecture" helped us to encapsulate the choice of appropriate strategy from the way in which any specific tailoring action is carried out. To continue the example from above, the virtual observation team can be used to scaffold the learner and to make the situation more complex. For example, the virtual team can report an event that serves as a distraction from the learner (attempting to take attention from events of the primary narrative) or misinforms (attributing a reported action to the wrong actor). These different tailoring options are mapped to individual learning objectives (e.g., "maintaining situational awareness") but are not directly connected to a specific instructional approach. As a consequence, the system can select and execute a tailoring option like "report a distracting event" under many different instructional contexts and student proficiencies. This design approach, which is deliberately modeled on Soar's separation of operator proposals and operator applications, provides significant flexibility in choosing what tailoring options are appropriate for some specific learner at some specific point in the learning trajectory.

Relatedly, we also plan to use the selected instructional strategy to modulate the impact of student actions on proficiency estimation. The notion is that positive evidence in highly scaffolded situations should be discounted in comparison to similar evidence without scaffolding. Similarly, negative evidence of proficiency under highly challenging circumstances may also need to be discounted. Thus, instructional and practice contexts will be considered together in order to gauge performance and proficiency more effectively. The cognitive systems approach we have taken is making this contextual interpretation challenge more manageable and scalable and reusable across domains.

3.2.4 Providing Richer System Understanding of Learner State

Dynamic tailoring can be targeted to address the affective states of a learner, as well as domain content. Recent evidence suggests affective states, rather than a secondary concern, are comparable in importance to the resulting learning outcomes as the design of learning content itself (Woolf et al., 2009). Further, understanding learner affective states is especially important when a goal of the practice environment is to cause (manageable) overloads on attention and stress to attempt to suggest real-world conditions.

In keeping with these findings, the dynamic tailoring implementation includes tailoring strategies that employ narrative adaptation and user-interface manipulations to attempt to engage and to increase/decrease stress. For example, we took advantage of the turn-taking game mechanic of the Cultural Meeting Trainer to surprise a learner by having the learner's conversational partner sometimes make a statement "out of turn" when the learner was inattentive. The specific utterances of the character are chosen based on the affective goal and the learner's level of proficiency in the current stage of the meeting. For a novice learner who was observed to be inattentive, the character could interrupt with a benign utterance such as "May I offer you a cup of tea?" For more advanced students, the character's interruption was more direct and referenced the inattention directly: "I see you have other things on your mind. Should we continue this meeting?" This latter statement could be coupled with an angry expression or aggressive posture.

This type of tailoring strategy is predicated on being able to assess the affective state of the learner. A range of sensors (e.g., EEG, ECG, eye tracking) can provide real-time indicators of a learner's cognitive state to enhance diagnostics and enable better tailoring (Reeves, Schmorrow, & Stanney, 2007; Woolf, et al., 2009). Explorations to-date use commercial EEG and a combination of psychophysiological sensors (ECG, EDR) to dynamically measure attention/arousal to support this strategy.

The challenge from a cognitive systems perspective is to incorporate input from a range of sensors in a more principled and reusable manner than the purpose-built interfaces we have so far implemented. The current approach, as suggested in Figure 1, is a pre-processing and fusion component, the Learner Instrumentation Middleware. This component is designed to normalize and fuse individual sensor input channels into a high-level symbolic representation that can then be accessed by the Monitor and other components (Wray, Folsom-Kovarik, & Woods, 2013). The middleware design includes a semantic transform layer that brings to bear expectations about recent and current student activity from the Monitor to facilitate disambiguation. In other words, the process is designed to use the learning context as interpreted by other DTS components to simplify the fusion process. However, full-scale implementation and an assessment of the actual performance and utility of this capability have not yet been initiated.

4. Summary and Conclusions

This paper presented an approach to dynamic adaption of learner practice implemented on a cognitive architecture foundation. We described how the challenge of dynamic tailoring, based on its requirements, would likely benefit from an integrated cognitive systems approach and then illustrated, in the form of review of one implementation, some of the capabilities and benefits provided by the cognitive architecture. The cognitive architecture helps meet the original functional requirements for tailoring and new ones as they have arisen in researching dynamic tailoring and extending the software capability to new applications and domains.

We have made significant progress in building a general dynamic tailoring capability and we illustrated with several example domains. Indeed, an advantage of the integrated cognitive systems approach, in comparison to solutions purpose-built for individual domains, is the reuse and reapplication of the underlying capabilities of the cognitive architecture as well as the knowledge components and algorithms realized via those fixed mechanisms and representations.

Of course, cognitive architectures are themselves a subject of research and subsequent extension and refinement of the cognitive-architecture level. Since we began developing the dynamic tailoring system in Soar (version 8), Soar has undergone a major extension, Soar 9, that added new memories and learning mechanisms (Laird, 2012). We anticipate that migration of the Dynamic Tailoring System to Soar 9 would provide support in meeting existing requirements more efficiently and/or inexpensively and could facilitate addressing new requirements. A few examples include:

- Using Soar 9's semantic memory to encode declarative knowledge used in the expert and assessment models, which would further reduce working memory size and overall memory footprint;
- Using Soar 9's episodic memory to record learner actions and assessments, which will not only serve to reduce the size of working memory (as in Section 3.2.2) but could also enable access to fine-grained histories of learner behavior, which could in turn be used to more systematically vary learner experience and identify patterns of behavior across learners; and
- Using Soar 9's reinforcement learning to tune the performance of individual components and reduce knowledge engineering requirements, and potentially to learn contextually-mediated proficiency models (as described in 3.2.3).

Although there is a familiar tension in software development in deciding when to migrate to a newer version of the architecture, the important point is that these new capabilities and enhancements can be readily envisioned given the new version of the architecture. The new architecture thus shapes and guides the future development of the application.

Acknowledgements

This work is supported in part by the Office of Naval Research. 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. This work was supported, in part, by the Office of Naval Research projects N00014-11-C-0193, Perceptual Training Systems and Tools (PercepTS) and N00014-1-C-0170, Tactical Semi-Automated Forces for Live, Virtual, and Constructive Training (TACSAF). We would also like to thank colleagues who have contributed directly to the development of the conceptual design and implementation of the DTS including Ben Bachelor, Richard Clark, Mark Core, Jeremiah Folsom-Kovarik, Eric Forbell, Josh Haley, Laura Hamel, Lisa Holt, Robert Hubal, Randolph Jones, Keith Knudsen, H Chad Lane, Brian Magerko, Robert Marinier, Charles Newton, Brian Stensrud, Adam Sypniewski, Jim Thomas, and Michael van Lent.

References

Anderson, J. A., Corbett, A. T., Koedinger, K., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences, 4*(2), 167-207.

Brown, J. S., & Van Lehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science, 4*, 379-426.

Clark, R. E. (2009). How much and what type of guidance is optimal for learning from instruction? In S. Tobias & T. M. Duffy (Eds.), *Constructivist Instruction: Success or Failure?* New York: Routledge, Taylor and Francis

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-493).

Corbett, A. T., & Anderson, J. R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User Adapted Interaction, 4*, 253-278.

Davis, E., & Miyake, N. (2004). Explorations of Scaffolding in Complex Classroom Systems. *The Journal of the Learning Sciences, 13*(3), 265-272.

Fletcher, J. D. (2009). Education and Training Technology in the Military. *Science*, *323*(5910), 72-75.

Fletcher, J. D., & Chatham, R. E. (2010). Measuring Return on Investment in Military Training and Human Performance. In P. E. O'Connor & J. E. Cohn (Eds.), *Human Performance Enhancements In High-Risk Environments* (pp. 106-128). Santa Barbara, CA: Praeger/ABC-CLIO.

Fournier-Viger, P., Nkambou, R., & Mephu, E. (2010). Building Intelligent Tutoring Systems for Ill-Defined Domains. In R. Nkambou, R. Mizoguchi & J. Bourdea (Eds.), *Advances in Intelligent Tutoring Systems* (pp. 81-101). Berlin: Springer-Verlag.

Heeter, C., & Winn, B. (2008). Implications of gender, player type and learning strategies for the design of games for learning. In Y. Kafai, C. Heeter, J. Denner & J. Sun (Eds.), *Beyond Barbie and Mortal Kombat: New perspectives on gender and gaming*. Cambridge, MA: MIT Press.

Katz, S., Lesgold, A., Hughes, E., Peters, D., Eggan, G., Gordin, M., et al. (1998). Sherlock 2: An intelligent tutoring system built on the LRDC framework. In C. P. Bloom & R. B. Loftin (Eds.), *Facilitating the development and use of interactive learning environments*. Mahwah, NJ: Erlbaum.

Laird, J. E. (2012). The Soar Cognitive Architecture. Cambridge, MA: MIT Press.

Laird, J. E., & Derbinsky, N. (2009). *A Year of Episodic Mempory*. Paper presented at the IJCAI 2009 Workshop on Grand Challenges for Reasoning from Experiences.

Laird, J. E., Derbinsky, N., & Voigt, J. (2011). Performance Evaluation of Declarative Memory Systems in Soar *Proceedings of the 20th Behavior Representation in Modeling & Simulation Conference* (pp. 33-40). Sundance, UT.

Laird, J. E., & Wray, R. E. (2010). *Cognitive Architecture Requirements for Achieving Artificial General Intelligence*. Paper presented at the Proceedings of the Third Conference on Artificial General Intelligence (AGI)

Lane, H. C., & Johnson, W. L. (2008). Intelligent Tutoring and Pedagogical Experience Manipulation in Virtual Learning Environments. In J. Cohn, D. Nicholson & D. Schmorrow (Eds.), *The PSI Handbook of Virtual Environments for Training and Education* (Vol. 3). Westport, CT: Praeger Security International.

Magerko, B. (2007). Evaluating Preemptive Story Direction in the Interactive Drama Architecture. *Journal of Game Development, 3*.

Magerko, B., Stensrud, B., & Holt, L. S. (2006). *Bringing the Schoolhouse Inside the Box - A Tool for Engaging, Individualized Training.* Paper presented at the 25th Army Science Conference, Orlando, FL.

National Academy of Engineering. (2008). *Grand Challenges for Engineering*. Washington, DC: National Academy of Sciences/National Academy of Engineering.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, Massachusetts: Harvard University Press.

Ogan, A., Wylie, R., & Walker, E. (2006). *The challenges in adapting traditional techniques for modeling student behavior in ill-defined domains*. Paper presented at the Workshop Proceedings on Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems.

Pea, R. D. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *The Journal of the Learning Sciences, 13*(3), 423-451.

Puntambekar, S., & Hübscher, R. (2005). Tools for Scaffolding Students in a Complex Learning Environment: What Have We Gained and What Have We Missed? *Educational Psychologist, 40*(1), 1-12.

Riedl, M. O., Lane, H. C., Hill, R., & Swartout, W. (2005). *Automated story direction and intelligent tutoring: Towards a unifying architecture.* Paper presented at the Papers of the Narrative Learning Environments Workshop at the 12th International Conference on Artificial Intelligence in Education.

Schatz, S., Reitz, E., Nicholson, D., & Fautua, D. (2010). *Expanding Combat Hunter: The science and metrics of Border Hunter*. Paper presented at the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC).

Schatz, S., Wray, R., Folsom-Kovarik, J. T., & Nicholson, D. (2012). *Adaptive Perceptual Training in a Virtual Environment*. Paper presented at the Human Factors and Ergonomic Systems (HFES-2012).

Tambe, M., Newell, A., & Rosenbloom, P. S. (1990). The problem of expensive chunks and its solution by restricting expressiveness. *Machine Learning, 5*, 299-348.

Thomas, J., & Young, R. M. (2011). Annie: Automated Generation of Adaptive Learner Guidance For Fun Serious Games. *IEEE Transactions on Learning Technologies, 3*(4), 329-343.

Tobias, S., & Duffy, T. M. (Eds.). (2009). *Constructivist Instruction: Success or Failure?* New York: Routledge, Taylor and Francis.

Van Lehn, K. (2006). The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

Van Lehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist, 46*(4), 97-221.

Van Lehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., et al. (2005). The Andes Physics Tutoring System: Lessons Learned. *Int. J. Artif. Intell. Ed., 15*(3), 147-204.

Vosniadou, S., Skopeliti, I., & Gerakaki, S. L. (2007). Understanding the role of analogies in restructuring processes. In A. Schwering, U. Krumnack, K. U. Kuhnberger & H. Gust (Eds.), *Analogies: Integrating multiple cognitive capabilities*. Osnabruck, Germany: Institute of Cognitive Science.

Vygotsky, L. S. (1978). *Mind and society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Woolf, B. P., Burleson, W., Arroyo, I., Dragon, T., Cooper, D., & Picard, R. (2009). Affect-aware tutors: recognising and responding to student affect. *Int. J. Learning Technology, 3*(3/4), 129-163.

Wray, R. E., Folsom-Kovarik, J. T., & Woods, A. (2013). Instrumenting a Perceptual Training Environment to Support Dynamic Tailoring. In D. Schmorrow & C. Fidopiastis (Eds.), *Proceedings of 2013 Augmented Cognition Conference*. Las Vegas: Springer-Verlag (Lecture Notes in Computer Science).

Wray, R. E., & Jones, R. M. (2005). An Introduction to Soar as an Agent Architecture. In R. Sun (Ed.), *Cognition and Multi-agent Interaction: From Cognitive Modeling to Social Simulation* (pp. 53-78). Cambridge, UK: Cambridge University Press.

Wray, R. E., Lane, H. C., Stensrud, B., Core, M., Hamel, L., & Forbell, E. (2009). *Pedagogical experience manipulation for cultural learning*. Paper presented at the Workshop on Culturally-Aware Tutoring Systems at the AI in Education Conference

Wray, R. E., & Munro, A. (2012). *Simulation2Instruction: Using Simulation In All Phases of Instruction*. Paper presented at the 2012 Interservice/Industry Training, Simulation, and Education Conference.

Wray, R. E., Woods, A., & Priest, H. (2012). *Applying Gaming Principles to Support Evidence-based Instructional Design*. Paper presented at the 2012 Interservice/Industry Training, Simulation, and Education Conference.