Lessons from Building Diverse Adaptive Instructional Systems (AIS)

Eric Domeshek, Sowmya Ramachandran, Randy Jensen, Jeremy Ludwig, Jim Ong, and Dick Stottler

Stottler Henke Associates, Inc.,
1650 S. Amphlett Blvd., Suite 300
San Mateo, CA 94402, USA
domeshek@stottlerhenke.com

Abstract. This paper presents lessons learned from building a wide range of Adaptive Instructional Systems (AISs), ultimately bearing on the question of how to characterize the space of potential AISs to advance the cause of standardization and reuse. The AISs we consider support *coached practice of complex decision-making skills*—e.g., military tactical decision-making, situation assessment, and systems troubleshooting and management. We illustrate forces that affect system design and dimensions along which systems then vary.

The relevant forces derive from the AIS's area of application, the project structure within which it is built, and the customer's priorities. Factors to consider include (1) The extent to which the target domain is well-defined versus ill-defined; (2) The degree of fidelity required, preferred, and/or available for an exercise simulation environment; (3) The intended roles of automated and human instructors in instructional delivery; and (4) Imperatives for short-term and/or long-term cost containment.

The primary dimensions of AIS design we consider in this paper include (1) Exercise Environment; (2) Knowledge Models; (3) Tutor Adaptations; and (4) Supporting Tools. Each of these is further broken down into a set of more detailed concerns. Together, they suggest structures that can inform an ontology of AIS methods and modules.

Keywords: Adaptive Instructional Systems, Application and Project Demands, Design and Implementation Choices, Ontologies and Standardization

1 Experience with Intelligent Tutoring Systems (ITSs)

Funded primarily by DoD and NASA, Stottler Henke has developed dozens of Intelligent Tutoring System (ITSs) for training and education in a wide range of subject areas. By reflecting on our direct experience, this paper aims to describe patterns that may help advance broader standardization goals for future applications that share features with some of our past ITSs. Most of these systems support coached practice of complex decision-making skills, such as military tactical decision-making, situation assessment, or systems troubleshooting and management. To support coached practice, ITSs typically integrate interactive exercise environments—e.g., simulations or problem-solving tools—that present problems, tasks, or scenarios to the student (Woolf, 2010).

Training exercises typically correspond to activities performed on the job. *Educational* exercises may teach knowledge and skills that are less tightly coupled to real-world tasks. In either case, the student works in the environment to perform task-relevant actions observable by the automated tutor. Students may also provide additional evidence of their thought process when analyzing situations or making decisions. The tutor may communicate with the student directly, via user interface widgets or avatar, or it may communicate indirectly via simulated characters.

Stottler Henke ITSs, like other tutoring systems, adapt their instruction to individual students by:

- Selecting exercises and instruction to address an individual student's current learning needs, and modifying the course of exercises based on student performance; and
- Providing individually tailored interventions such as prompts, hints, performance feedback, questions, and remedial instruction, before, during, and after exercises.

The applications our ITSs address, the project structures within which we build them, and our customers' priorities greatly influence the methods and technologies we apply. Consider some examples of these kinds of constraints:

- **Well- Versus Ill-Defined Domains:** In *well-defined domains*, common ITS design approaches involve tracing fully competent expert cognitive performance models, often accompanied by models of flawed student performance [1,2,3]. Such ITSs can assess performance by matching the student's actions with correct and buggy solutions generated from these models. It is feasible (though challenging and costly) to develop a sufficiently complete performance model when a subject is well-bounded and stable over time, such as for high school math or physics. However, many of the ITSs we have developed address *ill-defined domains* [4,5], involving complex decision-making in poorly bounded domains, in which the knowledge and skills—and therefore the exercises that teach them—must keep up with changing situations, methods, and tools used in the real world. Thus, we often implement ITSs that complement limited general-purpose performance models with more complex scenario-specific models of expertise.
- **Fidelity of Simulated Task Environments:** Rooted in the doctrine, "train as you fight," DoD customers often have a strong preference for high fidelity simulation. Unfortunately, using a high-fidelity simulation often incurs high costs, whether the ITS includes a new simulator or integrates with a legacy system [6,7]. Instrumenting legacy simulations so they report the kinds of information needed by an ITS is a recurring problem; as is finding means to route ITS output into the user interaction context created by a legacy simulation [8]. Alternately, quickly constructing a "cognitively realistic" simulation offers its own challenges, especially when it must include appropriately responsive non-student agents.
- **The Role of Instructors:** In our experience, practical ITSs are often used in blended learning environments, so most ITSs should also help human instructors understand and address the needs of individual students and the class overall.
- **Cost Containment:** There are always constraints on the time and money available to build an ITS: its exercise environment and instrumentation, its control knowledge and logic, and its instructional and exercise content. Since training applications tend

to change relatively quickly, the systems often need to be modified and maintained over time—new doctrine accommodated, new exercises created, etc.—ideally without having to establish another contract and pay for expensive AI expertise on an ongoing basis. This drives a desire for *authoring tools*, which in turn affects the forms that the ITS's knowledge and supported algorithms can practically take [9,10].

The overarching goals of suiting the task domain, addressing the learning objectives, and achieving training effectiveness within available resource limitations can be addressed in many ways. Thus, as illustrated below, we have employed a variety of approaches to developing adaptive exercise environments, assessment knowledge, and instructional interactions.

2 Requirements and Dimensions for AISs

In order to adapt instruction, one must be able to:

- Assess the student's knowledge and skills (and possibly attitudes, aptitudes, and emotional states), based on performance within the work environment and/or interactions with the tutor, perhaps supplemented by legacy instruments; and
- Select and present instructional interactions at the right time and in an effective manner to address issues identified during assessment.

In the context of the broader set of issues raised by ITSs, we have found it useful to characterize our systems along the following dimensions, all of which affect adaptation:

1. Exercise Environment
   a. Degree of Free Play
   b. Modes of Interaction
   c. Content of Interaction
   d. Interpretation Issues
   e. Simulation Constraints
2. Knowledge Models
   a. Expert Model
   b. Tutor Model
   c. Student Model
3. Tutor Adaptations
   a. Instruction Selection
   b. Exercise Selection
   c. Prompting
   d. Hinting
   e. Immediate Feedback
   f. After-Action Review
   g. Socratic Dialog
   h. Remedial Instruction
4. Supporting Tools
   a. Instructor Support
   b. Authoring Support

In the following section, we provide discussion and examples of these dimensions.

3 System Design Dimensions and their Effects on Adaptation

3.1 Exercise Environment

Degree of Free Play. Degree of Free Play [11] is our first environmental factor. Free play, here, means relatively unconstrained—yet still tracked and tutored—pursuit of specific training-defined tasks. It does not imply relatively undirected exploration of a simulated "microworld" [12,13]. Consider two possible points on this dimension and how they affect adaptation.

- Some of our systems such as the Tactical Action Officer (TAO) ITS [14,15] provide free-play simulations whose events evolve realistically in response to a wide range of possible student actions. The timing and sequence of tactical situations depends upon actions taken by computer-generated forces, which in turn respond to student actions and the actions of other simulated agents. Thus, one cannot assess student performance simply by recognizing student actions at prespecified times or in a specified sequence. Instead, student actions must be evaluated in the context of the tactical situation by considering the state of other friendly and opposing forces and their recent actions. Thus, the assessment subsystem within TAO-ITS uses Behavior Transition Net-works (BTNs), an extension of finite state machines. Developed at Stottler Henke, BTNs detect significant sequences of events, conditions, and student actions to assess performance and identify knowledge and skill gaps [16,17].
- By contrast, effective training systems can often be created that provide semi-free play simulations in which a moderate number of options are made available to the student. For these kinds of training simulations, we have developed an in-house intelligent tutoring engine called the Task Tutor Toolkit (T3) [18, 19]. T3 matches student actions to a solution template that encodes correct actions within a scenario and their allowable variation. Although this approach supports simulations that are somewhat constrained, compared to free-play simulations, the development of these simulations and the solution template for each scenario requires much less effort.

Modes of Interaction. The available forms of input and output (I/O) supported by an exercise environment are a major determinant of training realism, a potential cost-driver of ITS development, and an opening for widening the range of adaptations the system can exhibit. Most of our systems include traditional graphical user interface (GUI) elements with forms and widgets, often designed to mimic or abstract some real-world equivalent—e.g., systems control interfaces or traditional paperwork. Many of our systems include more strongly graphical elements such as maps (for tactical situations as in TAO-ITS, InGEAR [20], ComMentor [21,22], and many others), diagrams (for complex logic or control systems as in ICT Tutor [23]), or graphs (for data analysis problems as in AAITS [24]). We have done some work on leveraging 3D or virtual reality environments, including Commercial Off-The-Shelf (COTS) game engines. We have devoted more effort to allowing for language-based interaction, including speech-based I/O. For instance, students using the second generation TAO-ITS system could converse with simulated watchstanders staffing a ship's combat center, and students in METTLE [25] could engage in a diagnostic interview with a simulated patient. In the line of prototypes starting with ComMentor, we explored mechanisms to support extended Socratic dialogs.

No matter the external format, a system can ultimately only respond sensibly to some range of inputs. In some situations, widgets are used to build tightly restricted forms of input, such as multiple-choice questions or checklists. In other situations, we use more flexible mechanisms, such as interactive graphics (e.g., diagrams, maps, and timelines), or constrained understanding and generation of spoken language and typed text. Most diagrams have some natural structure—e.g., nodes and/or links with some sets of available states—that effectively defines what can be communicated. Many maps or data presentations can be analyzed into a smaller set of qualitatively distinct and meaningful regions. Even with language, a massive universe of student utterances can be mapped to a much smaller set of expected and meaningful inputs. Driven by domain needs and project constraints, the recognizable inputs may constitute a relatively large or quite small set of alternatives to which the exercise environment and/or tutor most adapt. We will say more about the kinds of content that can be communicated and the degree of contextual interpretation required even once a student input is recognized/classified.

Specifying and controlling a range of environment or tutor outputs presents a different and often simpler set of challenges. For instance, understandable language generation is easier than language interpretation. Utterances can be completely canned or templated to allow some useful range of variation. In some cases, pre-recorded audio and video are appropriate. Means to inject language or other kinds of environment manipulations into an exercise can be designed into custom-built simulations. However, introducing ITS-driven adaptive behaviors into an existing simulation can be more of a challenge.

Content of Interaction. A typical exercise environment focuses on providing means to support carrying out a task to be trained. However, sometimes, it is not possible to assess the student's knowledge and skills based solely on their observable task-relevant actions. For such applications, it is useful to elicit the student's reasoning, either by requiring the student to show their reasoning as part of the assigned task, or by asking the student questions about their thought process. As an example of showing reasoning, the ITADS [26, 28] tutor provides a "rationale panel" where students can maintain an inventory of hypotheses underlying their actions in its free play diagnostic environment. As an example of asking about thought processes, the ComMentor system presents tactical decision games that prompt students, typically Army Captains, to sketch tactical plans. Then, the system evaluates the student's plan and engages in a Socratic dialog that probes their understanding of the situation, guiding them through questions and feedback to build up an argument structure for appropriate interpretations and courses of action. AAITS teaches underwater acoustic analysis and the ICT Tutor teaches counter-intelligence analysis. Both of these systems require students to enter observations and inferences, using a domain-specific, graphical analysis tool. In all these cases, the tutors condition their assessments and their instructional support not just on overt task performance, but also on revealed student rationale.

Interpretation Issues. Student inputs often require contextually sensitive interpretation in order to support tracking and assessment of performance and learning. This context-sensitivity can exist at different points on the free play spectrum, and for different input forms and contents, even after basic input meaning is determined. As noted earlier, for free play exercises as in TAO-ITS, we use BTNs to evaluate student actions in their tactical context, including the state and recent actions of other friendly and opposing forces. For constrained play exercises using T3, student actions are interpreted with respect to their place in an observed sequence relative to a flexible solution template. In ITADS student assertions about hypotheses are compared to the system's own record of justifiable hypotheses given the diagnostic information uncovered by student actions. In ComMentor, student input about their understanding of the situation and decision rationale are interpreted relative to a set of argumentative points to be discussed, while tracking context for topics that have already been raised, are currently-in-focus, and are yet to-be-discussed. Without context-sensitive interpretation, tutor assessments will often be incorrect and system adaptations ill-informed.

Simulation Constraints. An ITS's exercise environment may be custom-built, rely on integration with a legacy system, or use some combination of the two. Beyond the cost and realism impacts, the *build* and *reuse* options tend to introduce different constraints on observing student activity and controlling the student experience—e.g., modifying the course of the exercise and/or injecting tutor interactions. For instance, the second-generation PORTS/TAO-ITS was integrated with the legacy Aegis PORTS simulator; enabling the ITS BTNs to observe and manipulate elements of the simulation required close collaboration with the PORTS developers. Similar collaboration was required for InGEAR's integration with a legacy game. Sometimes, however, opportunities to ask for access to desired interfaces in legacy systems may be limited, as when we used the standard High Level Architecture (HLA) interfaces for the BC2010 ITS [26]. For ITADS, we were required to provide a high-fidelity simulation of a representative shipboard information technology environment. The resulting network of custom-configured virtual machines required new instrumentation in order to support rich observation and control. For the students' free play troubleshooting task in ITADS, we were forced to invent new forms of modeling to maximize inferences given limited available observability of student actions. Within the same system, students also have a procedure execution task, for which we fell back on (more restricted) solution templates (like T3) for modeling and coaching. In contrast, systems such as AAITS, ICT, and METTLE all relied on custom-built exercise environments in which we could, with some development effort, make the environments do whatever was needed to support adaptive instruction.

3.2 Knowledge Models

In our ITSs, the dominant modeling effort is typically devoted to building the **Expert Model**—describing what constitutes good understanding and behavior in the domain, and therefore what we would like the student to learn. The details of the Expert Model are often abstracted and linked to an associated **Curriculum Model**—a hierarchy of principles that students are expected to master. The Curriculum Model summarizes *what* the students are to learn. The Expert Model contains all the details of *how* to recognize and/or perform adequate knowledge and behavior. Tutor interventions such as hints, prompts, and feedback, can be associated with either or both of Curriculum nodes and Expert behaviors. The **Tutor Model**—controlling delivery of available tutor interventions in response to student performance and states of the Student Model—is typically left as code, though often with some parameterization to allow fine-tuning to suit instructors' pedagogical preferences. The **Student Model** is typically a mastery overlay on the nodes of the Curriculum Model, capturing the results of automated assessments that are part of the Expert Model.

The discussion here will focus on the various forms the Expert Model takes in different ITSs, with the aim of achieving training effectiveness within available resource limitations while suiting the task domain and learning objectives.

- **Expert Behaviors:** Probably the most flexible form of Expert modeling we use is context-sensitive sequence recognizers of the kind most easily built using BTNs (as applied in TAO-ITS, the C2V-ITS [29, 30], and many other systems). An important issue with such models is the extent to which the resulting behavior specifications are situationally bound. That is, does a BTN apply only within a specific scenario (or some part of a scenario) or is it a more generally applicable characterization of good performance? It is common for our ITSs to include a combination of scenario-specific and cross-scenario knowledge. Versions of this question arise for other modeling approaches discussed below. Our SimBionic system for BTN authoring supports hierarchical decomposition and target-specific variants of BTNs, which can be useful when seeking to partition and generalize behavior specifications for reuse.
- **Domain Constraints:** Another powerful and common approach to modeling Expert knowledge and behavior is to capture constraints among domain objects and actions. This approach was used in a series of tactical tutors such as InGEAR, AAIRS [31, 32], and BC2010 ITS. InGEAR paid particular attention to generalizing such constraints by starting the process of formalizing key concepts to facilitate automated recognition in different situations—e.g., characterizing the concept of *cover and concealment* as it applies to different terrains. ITADS troubleshooting Expert Model exploited constraints on what symptoms provided evidence for or against what faults, and what student actions could produce additional evidence. The form of the model was envisioned as potentially applying across a wide range of troubleshooting applications.
- **Solution Templates:** The template approach introduced by T3 provides a lower-cost, though more limited, means to track and coach activity when more flexible approaches are not needed or perhaps not affordable. A similar scheme was used for those aspects of ITADS focused on fixing (rather than diagnosing) problems. Variation in activity ordering and parameterization, as well as some context-sensitive selection of alternatives can be accommodated. Coaching can be provided to help keep the student making progress along a viable path. Again, templates can be written so they apply across multiple exercises.
- **Reusable Script Elements:** Scripted exercises seem like they would offer only limited adaptation. However, the degree of limitation depends on the sophistication of the scripting formalism and the size of the scripts. For example, the METTLE system is explicitly built on a concept of scripting, but the scripts are very large (e.g., hundreds of lines for a simulated patient) and each script line is conditional. Conditions can include not only immediate triggers—such as things just done or said by the student—but also tests depending on earlier actions or logical combinations of such actions. To speed authoring and promote generality, script lines can be reused by particular actors in specific scenarios and/or scenes, with selective overrides for chosen aspects of such lines—e.g., conditions, actions, or instructional annotations.

- **Direct Mastery Evidence:** In some ITSs, exercises are composed from more focused interaction—e.g., questions and available answers—that are directly linked as evidence for or against mastery of Curriculum principles or skills. For instance, in ReadInsight, a text passage is accompanied by (an adaptive set of) comprehension questions. Available answers to each question are treated as mastery evidence.

3.3 Tutor Adaptations

Across our many ITSs we have built versions of all the kinds of tutor adaptation listed in Section 2. **Instruction Selection** and **Remedial Instruction** are probably the two simplest techniques in our work. That is because instruction is generally taken to be some kind of mostly-opaque multimedia, often prepared and possibly delivered using external COTS tools such as PowerPoint, Articulate, or Flash. In our systems, such instruction packets are linked to curriculum nodes and may be further tagged in other ways—such as being particularly appropriate for initial or secondary exposure. Instruction or remediation is chosen based on student curriculum mastery estimates, and subject to tunable heuristic rules regarding issue such as score thresholds for offering remediation, and prerequisite structures for introducing new topics. **Exercise Selection** is generally also driven by links to curriculum nodes and student mastery estimates. However, as noted in connection with TAO-ITS, since we control the exercises more completely, we can sometimes use *exercise configuration* or real-time *exercise steering* as alternatives to selecting an entirely new exercise.

During-exercise coaching includes **Prompting**, **Hinting**, and **Immediate Feedback**. Of these, prompting—taken to mean proactive tutor suggestions—is probably used least often. While most of our expert modeling approaches can determine some reasonable next action at most moments, it is harder to judge when it would be productive to break in on a student and suggest such a move. METTLE does this using author-settable timeouts that start running when some conditions are first met. Tactical decision-making tutors such as InGEAR and C2V-ITS monitor the passage of time as part of the tactical situation, such that the absence of student action can be a trigger for proactive suggestions. More often, the tutor's knowledge of useful next steps is used to drive a series of hint offerings. The student must explicitly request such hints (though the tutor may flag their availability). A common practice is to offer a progression of hints going from general to more specific—e.g., *what to consider, what to do, how to do it, why to do it*. Immediate feedback is typically tied to assessment logic, though again with additional control annotations and/or heuristics to determine which assessments are worth commenting about in the middle of the exercise flow. It is usually judged more important to provide *negative feedback* aimed at immediately correcting a student mistake—whether that be for pedagogical purposes, or to ease the system's student tracking chores—than to provide *positive feedback* reinforcing correct decisions. All of these interventions are generally under author control, since instructors may have opinions about what kinds of interventions to include. For instance, some military instructors may be less concerned about student motivation and hence de-emphasize positive feedback. All of these interventions can potentially be combined with *fading*, wherein control logic takes into account the student's assessed mastery level on relevant curriculum points when deciding whether or not to offer a prompt, hint, or feedback. Most of our systems focus exclusively on adapting based on mastery assessments. Again, military training addresses students who have been pre-selected for certain attributes and personality traits, so adaptation to factors beyond performance, such as motivation, may not be as interesting in that context.

After Action Review is provided by most of our ITSs. In its simplest form, it is a comprehensive collection of exercise assessments and paired feedback, like a report-card, organized by a scheme such as exercise chronology or curriculum hierarchy. Links to chosen remediations are typically embedded with the (negative) assessments. But assembling a truly effective AAR can require substantial adaptation, as illustrated by AAIRS. In a tactical trainer, making a point clearly can involve constructing an adaptive presentation that selects particular events, and then uses overlay graphics, vantage points, and filtering to highlight important information. Also, with team trainers, part of the challenge is to illustrate team dynamics and focus on individuals or sub-teams and their roles in the problem. If the tutor is not smart about how the AAR is assembled and just presents a (non-tailored) playback to the training audience for each training point, the benefits of individualized assessment may largely be lost.

Finally, **Socratic Dialogs** are a kind of extended structured interaction that can be used either as an element within AAR (as in ITADS), as an occasional reflective interlude (as in METTLE), or as the dominant format for an entire exercise (as in Com-Mentor). The surface form of a Socratic dialog is an exchange driven by tutor-generated questions. The intent of a Socratic dialog is to engage the student in guided self-explanation, and to provide relatively direct evidence to the tutor of mental processes, along with possible gaps and misconceptions. Thus, a Socratic dialog has both a teaching purpose and an assessment purpose. Our dominant approach—derived from Com-Mentor—uses a tree-structured argument script as the backbone for the interaction. Tutor questions aim to elicit key statements about the situation and/or proposed solution from the student. Any points the student misses or only partially states can be revisited through a combination of repeated focused questions, drill-down to pieces of an extended line of argumentation intended to build up to the missing insight, and/or tutor summary or recapitulation of the argument and supported point.

3.4 Supporting Tools

The need to provide **Instructor Support** has relatively minor impacts on adaptation. The main constraints it introduces are (1) curriculum elements must be meaningful and the hierarchy's organization comprehensible to an instructor (both strongly preferable in any case), and (2) connection of curriculum node mastery assessments to exercise behavior assessments be clearly traceable and explicable.

The need to provide **Authoring Support** has much more pervasive impacts on the forms of adaptation that a system can support. Generally, what is desired is a fully integrated authoring environment that supports the creation of, and linking among, curriculum, instruction, exercises, assessments, tutor interventions, etc. The greatest challenge is usually support for authoring of the Expert Model, in any of the forms surveyed in Section 3.2. The SimBionic toolkit provides robust support for drawing BTN logic as hierarchical flow charts. The T3 tools support authoring of more restrictive solution templates by demonstration of an activity sequence, followed, if necessary, by GUI-based editing to relax and add constraints or alternate paths. We have experimented with several schemes for authoring tree-structured Socratic dialogs, most recently embedded in ITADS and its authoring suite. Systems that exploit domain constraints typically introduce their own custom GUI editors with forms tuned to the formats of those constraints. That is true, for instance, with InGEAR and ITADS. Directly linking exercise actions as mastery evidence likewise calls for a custom editor, though relatively simple design suffice if users only need to link exercise options to curriculum elements. Finally, scripting schemes may depend on a custom structure editor or on a textual editor and parser for a scripting language.

As suggested by this paper, there are many degrees of freedom in ITS design (all the dimensions listed in Section 2). We have also suggested the need to make design tradeoffs, compromises, and innovations to suit domain and project constraints (all the examples in this Section 3). In our experience, those constraints are often discovered incrementally by working through examples. The consequence is that authoring tools are often unavailable and/or unreliable when early content needs to be developed because the formalism to be authored is still being invented and changed on the fly. This conundrum accentuates the value of highly polished tools that support proven families of modeling mechanisms, like SimBionic for BTNs and T3 for Solution Templates. We have explored other schemes to quickly bootstrap flexible, reliable, and helpful authoring tools. These include (1) mapping model structures into tabular formats and using COTS spreadsheet or database applications that support constrained input (such as Excel or Access); (2) mapping model structures into textual formats that are easily parsed (such as s-expressions or context-free grammars); (3) building tools using GUI frameworks that provide rapid configuration of custom editors driven by an underlying data model (such as the Eclipse Rich Client Platform and its Modeling Framework).

4 Towards an Ontology for AIS Design

AISs can be characterized in many ways—e.g., in terms of their domains of applicability, pedagogical commitments, styles of interaction, forms of modeling, bases for adaptation, ways of adapting, and so on. Our focus in this paper has been to point out how the combination of domain and project constraints can have pervasive impacts on the kinds of interaction, modeling, and adaptation that may be needed or possible, and, in turn, on the mechanisms that will work to provide those target capabilities.

Accordingly, when we think about carving up the space of AISs—especially as we look to identify and build reusable tools—we tend to think first in terms of how these *domain and project constraints point towards* <u>families</u> *of mutually compatible interaction, modeling, and adaptation mechanisms*. This is reminiscent of the points made in Bell [33] when he focused his breakdown of the AIS space around the questions: "how do I build one?" and "what's hard about that?" Our stance also shares features and some lineage with the approach taken within the Goal-Based Scenarios research program [34], where specialized authoring tools and runtimes were developed to support particular styles of interaction—in that case, centered on a major learner activity—and embedding guidance on how to construct pedagogically effective content.

The authors of this paper come to AIS design primarily with backgrounds in symbolic Artificial Intelligence. So, when it comes to developing ontologies to characterize AISs, we would naturally develop fine-grained breakdowns along any or all of the lines suggested above—domain attributes, pedagogical mechanisms, interaction styles, modeling approaches, adaptation drivers, and adaptation mechanisms. Each point in such an ontology space could suggest a potentially recurring situation or need or capability, and hence a possibly reusable technology or method or module. However, as illustrated in this paper, there will necessarily be dependencies across those different dimensions. Not every possible combination is likely to make sense, or be realizable, or fit with commonly recurring project constraints. We also suspect that most of these dimensions are somewhat open-ended or are at least likely to see further growth for some time.

Thus, we expect there could be a large number of reusable components, and that those components will vary in how commonly applicable and freely combinable they are. In consequence, we expect substantial advantage to pre-packaging consistent combinations or components, together tuned to address a range of recurring needs. Even if one overarching framework (say GIFT [35, 36]) could encompass configuration of all such modules, implementation effort would be reduced by re-using pre-configured packages that (nearly) addresses current needs. For example, a package containing expert model mechanisms tailored to troubleshooting domains could be constructed in a form that can be instantiated for future training applications involving troubleshooting skills. Further, some of the earlier work cited above notes that effectively exploiting the capabilities of reusable AIS modules or module-constellations remains challenging. That is, pedagogical effectiveness will ultimately depend on the how well the chosen pieces are used, so embedding authoring guidance is essential [33, 34].

Therefore, we would propose that in addition to fine-grained dimension-specific ontologies, AIS standardization would benefit from development of a higher-level catalog of AIS applications, characterized in terms of their use of the lower-level pieces. Such a catalog would be open-ended (non-exhaustive) and would need to allow for partial overlaps (non-exclusive). It would likely lead to introduction of additional lower-level taxonomies to characterize the roles and relationships of the primary components, as well as the applicability conditions of the modules and constellations.

Obviously, this implies a substantial community effort. Our range of experiences with coached practice for complex decision-making skills offers suggestive guidance, but substantial and ongoing input from the larger ITS and AIS community will be required to build and maintain a framework that more fully covers the full range of potential AIS requirements.

References

1. Anderson, J. R., Boyle, C. F., & Yost, G. (1985, August). The geometry tutor. In *IJCAI* (pp. 1-7).
2. Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The journal of the learning sciences*, *4*(2), 167-207.
3. Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., ... & Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, *15*(3), 147-204.

4. Jonassen, D. H. (1997). Instructional design models for well-structured and III-structured problem-solving learning outcomes. *Educational technology research and development*, *45*(1), 65-94.
5. Lynch, C., Ashley, K., Aleven, V., & Pinkwart, N. (2006, June). Defining ill-defined domains; a literature survey. In *Proceedings of the workshop on intelligent tutoring systems for ill-defined domains at the 8th international conference on intelligent tutoring systems* (pp. 1-10).
6. Fink, C. D., & Shriver, E. L. (1978). *Simulators for maintenance training: Some issues, problems and areas for future research*. KINTON INC ALEXANDRIA VA.
7. Andrews, D. H., Carroll, L. A., & Bell, H. H. (1995). The future of selective fidelity in training devices. *Educational Technology*, *35*(6), 32-36.
8. Stottler, R. H., Richards, R., & Spaulding, B. (2005, April). Use cases, requirements and a prototype standard for an Intelligent Tutoring System (ITS)/Simulation Interoperability Standard (I/SIS). In *Proceedings of the SISO 2005 Spring Simulation Interoperability Workshop* (pp. 3-8).
9. Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. In *Authoring tools for advanced technology learning environments* (pp. 491-544). Springer, Dordrecht.
10. Sottilare, R., Graesser, A., Hu, X., & Brawner, K. (Eds.). (2015). *Design recommendations for intelligent tutoring systems: authoring tools and expert modeling techniques*.
11. Andrews, D. H., & Windmueller, H. W. (1986). Lock-Step Vs. Free-Play Maintenance Training Devices: Definitions and Issues. *Educational Technology*, *26*(7), 29-33.
12. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.
13. Lawler, R. (1982). Designing computer based microworlds. *Byte, 7,* 138–146.
14. Stottler, R., & Harmon, N. (2001) Transitioning an ITS Developed for Schoolhouse Use to the Fleet: TAO ITS, a Case Study. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2001)*
15. Stottler, R., Davis, A., Panichas, S., & Treadwell, M. (2007) Designing and Implementing Intelligent Tutoring Instruction for Tactical Action Officers. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2007)*
16. Houlette, R., Fu, D., & Jensen, R. (2003) A Visual Environment for Rapid Behavior Definition. *Proceedings of the 2003 Conference on Behavior Representation in Modeling and Simulation, Scottsdale, AZ, 2003*
17. Presnell, B., Houlette, R., & Fu, D. (2007) Making Behavior Modeling Accessible to Non-Programmers: Challenges and Solutions. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2007)*
18. Ong, J., & Noneman, S. (2000) Intelligent Tutoring Systems for Procedural Task Training of Remote Payload Operations at NASA. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2000)*
19. Mohammed, J., Ong, J., Li, J., & Sorensen, H. (2005). Rapid development of scenario-based simulations and tutoring systems. In *AIAA Modeling and Simulation Technologies Conference and Exhibit* (p. 6419).
20. Jensen, R., Lunsford, J., Presnell, B., Cobb, M.G., & Kidd, D. (2013) Generalizing Automated Assessment of Small Unit Tactical Decision Making. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2013).* Paper
21. Domeshek, E., Holman, E., & Ross, K. (2002) Automated Socratic Tutors for High-level Command Skills. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2002)*

22. Domeshek, E., Holman, E., & Luperfoy. S. (2004) Discussion Control in an Automated Socratic Tutor. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2004)*
23. Ramachandran, S., Remolina, E., & Barksdale, C. (2006) Scenario-based Multi-Level Learning for Counterterrorism Intelligence Analysis. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2006)*
24. AAITS description, https://www.stottlerhenke.com/solutions/education-and-training/aaits-teaches-undersea-acoustic-analysis-to-navy-sonar-technicians/, last accessed 1/4/2019.
25. Domeshek, E. (2009) Scenario-Based Conversational Intelligent Tutoring Systems for Decision-Making Skills. *Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2009)*
26. Stottler, R. H., Pike, B., Bingham, R., and Jensen, R. (2002). "Adding an Intelligent Tutoring System to an existing tactical training simulation," Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2002).
27. Ramachandran, S., Domeshek, E., Jensen, R., & Aukamp, A. (2016) Uncovering the Hidden: Tradeoffs in Rationale Elicitation for Situated Tutors. *Proceedings of the Interservice / Industry Training, Simulation & Education Conference (I/ITSEC 2016)*
28. Ramachandran, S., Jensen, R., Ludwig, J., Domeshek, E., & Haines, T. (2018) ITADS: A Real-World Intelligent Tutor to Train Troubleshooting Skills. In *Artificial Intelligence in Education 19th International Conference*. Springer International Publishing
29. Jensen, R., Marshall, H., Stahl, J., and Stottler, R. (2003). "An Intelligent Tutoring System (ITS) for Future Combat Systems (FCS) Robotic Vehicle Command," Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2003).
30. Jensen, R., Mosley, J., Sanders, M., and Sims, J. (2009). "Intelligent Tutoring Methods for Optimizing Learning Outcomes with Embedded Training," Proceedings of the NATO Workshop on Human Dimensions in Embedded Virtual Simulation (NATO HFM-169).
31. Jensen, R., Nolan, M., and Chen, D.Y. (2005). "Automatic Causal Explanation Analysis for Combined Arms Training AAR," Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2005).
32. Jensen, R. (2009). "Assessing Perceived Truth Versus Ground Truth in After Action Review," Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2009).
33. Bell, B. (2015). One-size-fits-some: ITS genres and what they (should) tell us about authoring tools. *Design recommendations for intelligent tutoring systems*, *3*, 31-45.
34. Jona, M., & Kass, A. (1997). A Fully-Integrated Approach to Authoring Learning Environments: Case Studies and Lessons Learned. In *the Collected Papers from AAAI-97 Fall Symposium workshop Intelligent Tutoring System Authoring Tools. AAAI-Press.*
35. Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The generalized intelligent framework for tutoring (GIFT). *Orlando, FL: US Army Research Laboratory–Human Research & Engineering Directorate (ARL-HRED).*
36. Sottilare, R. A., Brawner, K. W., Sinatra, A. M., & Johnston, J. H. (2017). An updated concept for a Generalized Intelligent Framework for Tutoring (GIFT). *GIFTtutoring. org*.
37. Woolf, B. P. (2010). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.