Instrumenting a Perceptual Training Environment to Support Dynamic Tailoring

Robert E. Wray, Jeremiah T. Folsom-Kovarik, and Angela Woods

Soar Technology, Inc., 3600 Green Court Suite 600,
Ann Arbor, Michigan, USA, 48015
{wray,jeremiah.folsom-kovarik,angela.woods}@soartech.com

Abstract. Simulation-based practice environments would be more valuable for learning if they supported adaptive, targeted responses to students as they proceed thru the experiences afforded by the environment. However, many adaptation strategies require a richer interpretation of the student's actions and attitudes than is available thru the typical simulation interface. Further, creating extended interfaces for a single application solely to support adaptation is often cost-prohibitive. In response, we are developing "learner instrumentation middleware" that seeks to provide a generalized representation of learner state via reusable algorithms, design patterns, and software.

Keywords: Perceptual learning; adaptive training; learner modeling.

1 Introduction

Many of today's computer-based learning environments offer simulacrums of the performance environment. These practice environments enable a learner to practice skills and to demonstrate knowledge of concepts that are the subject of training, offering support for more sustained and thus potentially deeper and more complete learning [1-3]. Although theoretical debate continues regarding how to best structure practice experiences, an emerging consensus agrees that dynamic adaptation of practice to enable targeted, individualized experience is important for effective computer-based training [4].

We are taking an applied perspective to practice environments, focusing on delivering effective and adaptive instruction thru whatever means appears apt for the domain. Toward this end, we are developing general learner instrumentation and tailoring capabilities that enable practice environments to adapt to the learner both extrinsically (outside of the domain experience of the simulation) and intrinsically (within the simulated experience). These capabilities also are designed to support both learner cognitive and affective states. The resulting Dynamic Tailoring System [5, 6] is designed to integrate instructional methods and best practices as they are identified and validated and also serves as a testbed for researching such adaptation strategies.

The Dynamic Tailoring System (DTS) has been demonstrated in multiple practice domains. Each domain imposes specific requirements. Although many requirements

D.D. Schmorrow and C.M. Fidopiastis (Eds.): AC/HCII 2013, LNAI 8027, pp. 100–109, 2013.

are shared across domains, some are unique. A core engineering challenge is to define and implement a capability that is sufficiently functional and flexible to support the common and the unique requirements of a particular application. In this paper, we explore this tension, focusing on the general challenge of learner instrumentation: processing and packaging inputs from a learner and simulation to enable the classification/recognition of learner states. These states then help the system identify the best interventions for the individual student at that moment. To illustrate with a concrete example, we introduce a perceptual training application that demands more powerful instrumentation than earlier applications because the human practice task is primarily one of observation, in which explicit action, which is easier to recognize and assess, is relatively infrequent.

In response to these limitations, we are developing "learner instrumentation middleware" that transforms, supplements, and fuses simulation and learner data into a succinct representation of learner context and state. We describe requirements of the learner-instrumentation capability such as extensibility and reusability. We then describe how we are developing and applying this middleware in the context of the perceptual training application.

2 Enabling Dynamic Tailoring

We have been developing the Dynamic Tailoring System as a general-purpose software architecture for dynamic tailoring; that is, pedagogical experience manipulation during practice [1, 6]. The system is specifically designed to support many types of simulation-based training systems and domains. We have implemented a functional implementation of this system and have demonstrated it in multiple domains. Figure 1 illustrates the current architecture. There are three core functional components (boxes) and four primary representational components (database icons). Here, we briefly outline the overall design of the system in order to motivate and to provide context for the learner instrumentation challenge below.

Monitor. The Monitor observes learner actions, interprets those actions in the context of the learning situation (via a domain/expert model), assesses the learner's behavior in terms of active learning objectives, and then classifies the observed behavior using a behavior ontology. As we outline further below, the Monitor is supported by several translation layers ("learner instrumentation middleware") that decouple the details of simulation environments and learner sensing from the representations used for interpretation.

Pedagogical Manager. The Pedagogical Manager maintains an estimate of proficiency for each learning objective, decides between extrinsic mediation (such as an ITS dialog) and the intrinsic tailoring, and chooses alternative instructional strategies. For example, a "scaffold" tailoring strategy can be used when a learner has demonstrated high levels of competence but is transitioning to more complex challenges or new learning objectives within the domain [7]. The Pedagogical Manager also mediates choices between affective and domain-content tailoring strategies.

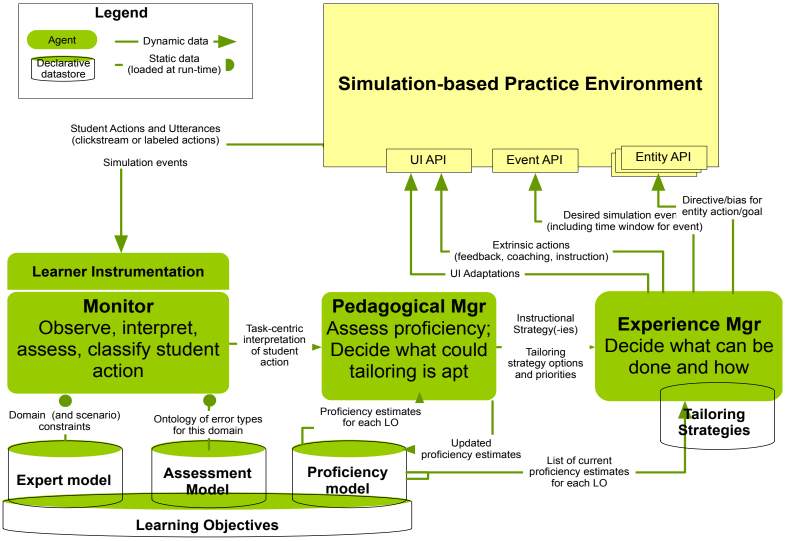

Fig. 1. Architectural composition of the Dynamic Tailoring System

Experience Manager. The Experience Manager chooses and instantiates tailoring strategies based on general recommendations from the Pedagogical Manager. For example, the Pedagogical Manager may recommend a tailoring strategy that is intrinsic, meant to challenge, and focuses on several enumerated learning objectives. The Experience Manager then evaluates tailoring strategy options to determine which strategies can be used to satisfy the request.

3 Instrumentation Requirements for Perceptual Training

The previous section outlined a high-level software architecture to support dynamic tailoring across a range of training applications. In this section, we wish to examine architectural requirements more substantively, focusing on a specific training application to highlight requirements. The training application includes a practice environment in which US Marines observe a village from a Virtual Observation Post (VOP).

The VOP is inspired and informed by successful “live” training programs [8, 9]. In this training, Marines learn to construct a general “baseline” of understanding from sustained attention to the activities in a “village” (populated by human role players). Marines exchange observations with one another and practice the application of observational skills introduced in a classroom. The resulting sensemaking skill covers a broad range of perceptual skills, from low-level signals (recognizing the proxemics and kinesics or “body language” of individual villagers), to recognizing and categorizing quotidian and unusual events, to developing an abstract mental representation of the patterns of life within the village.

For the VOP, the implementation of adaptive tailoring strategies is comparatively straightforward [10]. The learner is positioned in a Virtual Observation Post 1000m or more from the observed village. As a consequence, learner action requires less interaction and coordination with the simulation than in a domain in which a learner would be directly interacting with a business partner, an accident victim, or aircraft in a virtual battlespace. Learner actions include focusing "optics" (e.g., binoculars) on specific locations in the scene, reporting observations and events, and suggesting interpretations of events for others to consider and discuss. Open-ended speech recognition across a team of learners is a challenging technical problem, but, from the point-of-view of the tailoring system, these inputs are pre-processed to as labeled text strings.

Although learner actions are comparatively simpler, it is also relatively more challenging to develop and maintain an understanding of learning state in this domain. A learner may spend many minutes just scanning a scene with binoculars. During this time, numerous observations may be made (or missed) by the learner without an explicit utterance or formal report. An understanding of learner state is necessary for deciding what tailoring actions are relevant at a particular time for a particular learner/team. Without a good understanding of the learning state, appropriate and timely instructional tailoring is not possible. Worse, inapt tailoring may also increase learner frustration and negatively impact learning.

The requirements for tailoring in turn impose additional constraints on the practice environment. A practice environment without instructional supports (such as tailoring) needs only to allow a learner to take action in the environment (reporting an event to other team members, choosing different sensing optics, firing a weapon, etc.). Richer instrumentation of the practice environment is necessary to enable interpretation of a learner's actions and maintenance of a dynamic and reasonably accurate model of the learner.

Instrumentation is difficult because simulation affordances for learner observation are typically weak. Most simulations provide minimal descriptions of learner activity and without directly providing learner/task context; e.g., they may indicate that a

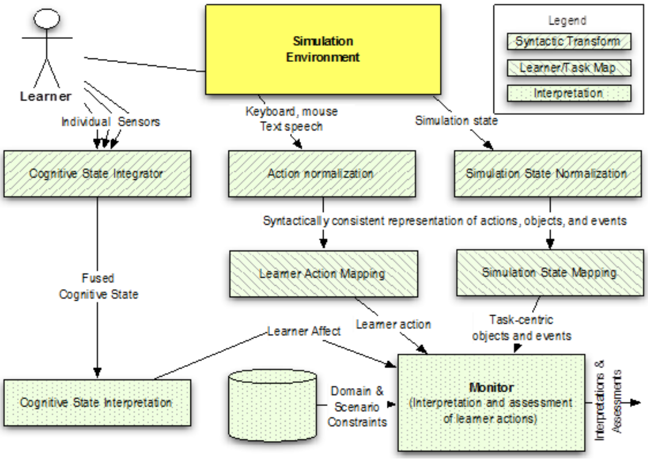

Fig. 2. Conceptual design of the learner instrumentation middleware

learner took some action at some time, but cannot indicate the appropriateness of that action. A related limitation is the available "channels" of observation. Most simulations provide only keystroke and mouse inputs, which are much more limited than the kinds of signals and information a human tutor might get from interaction with a learner.

4 Learner Instrumentation Middleware

In order to support adaptive training, the simulation environment requires additional methods of learner instrumentation in addition to just a practice environment. However, these requirements have the potential to add significant cost to the training system. The approach we have been investigating and exploring is "learner instrumentation middleware." This section outlines the design of this middleware, current progress toward the goal, and some of the tradeoffs in pursuing middleware versus application-specific solutions.

Fig. 1 illustrates the conceptual design of the learner instrumentation middleware. This software seeks to transform, supplement, and fuse simulation and learner data into a succinct representation of learner context and state with a general (not domain specific) set of functions and processes. The potential advantage of the middleware is that it can collect and transform individual sensor and input streams from the learner and simulation into representations that are largely independent of these sources. This approach allows the interpretation and adaptation algorithms used in the remainder of the Dynamic Tailoring System to be independent of the specific simulation environment and sensor suite.[1] There are three distinct layers to the middleware: 1) syntactic normalization, 2) learner/task mapping, and then 3) interpretation. Each of these layers is sketched individually below.

4.1 Syntactic Normalization

This layer converts the specific representations used by the simulation to a general representation used within the Dynamic Tailoring System. In the examples in this paper, we use a predicate representation of the normalized syntax for simplicity/clarity. This layer is largely custom-built for each simulation environment. However, this layer is not application specific, meaning that components in this layer can be reused for different training applications using a common simulation environment.

The Cognitive State Integrator (CSI) is a more sophisticated component than the other two in this layer. The goal of the CSI is to provide a consistent representation of estimated cognitive states regardless of the sensor(s) used to measure indices of these states. At this level of the middleware, indices measured from different sensors are fused and leveled, providing an estimate of a particular cognitive-state dimension (such as arousal or attention) at the current moment in time.

---

[1] An "inverse transform" is required for translation of adaptive interventions into simulation-specific functions.

4.2 Learner/Task Mapping

The role of this layer is to translate or map the outputs from the syntactic normalization layer to a representation that is focused on the learning task rather than simulation events. The transformation at this layer is in some respects analogous to an affine transformation, in that the transformation enables the consumer of the transformed data to be simpler. The mapping process simplifies and unifies the range of inputs the interpretation layer must consume, allowing interpretation to be domain neutral. Ideally, it will also be possible for the mapping layer to be reusable across domains. However, we have not yet developed general, reusable algorithms for this layer and it thus currently requires custom programming for each new application. We are currently investigating a scenario representation with a formal (ontological) representation within this layer to reduce and simplify the custom development requirements.

Several examples of the kinds of mapping provided in this layer are summarized in Fig. 3. Imagine a situation where a learner is tasked to track and report on the actions of an individual moving thru a small village. At some point, this actor enters a marketplace. The learner, using virtual binoculars with high magnification, notices that the high-interest individual is visibly angry and reports that over a simulated radio. The syntactic layer, as above, converts data from the different components of the simulation – an optics simulation, a simulation environment (e.g., VBS2), speech-to-test components – and converts them into to a predicate representation similar to that pictured in Fig. 3.

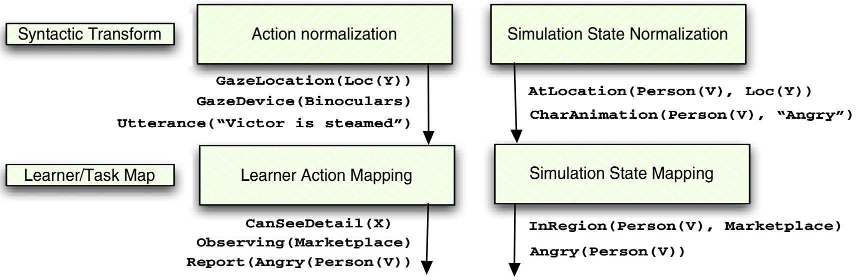

Fig. 3. An example of inputs and outputs for the Learner/Task Mapping layer

This translation decouples the syntactic details of the different system components for interpretation but representations remain tied to their original frames. For example, the syntactic layer will provide data about the location of particular objects, but not how those objects relate to the learner and the learning task. The mapping layer provides this translation. As suggested by the figure, the combination of the use of binoculars and focus on the marketplace allows the system to be able to infer that the learner `CanSeeDetail()`of objects in the market, such as the angry expressions of the high-interest individual. As we discuss further below, these mappings to the learning context make it much simpler for the interpretation layer to reason about the learning situation and assess a learner's action(s).

4.3 Interpretation and Assessment

Monitor. The syntactic and semantic transform layers feed the interpretation and assessment layer. As outlined above, the "Monitor" evaluates the current learning state, as represented by the outputs of the semantic transform layer against a collection of user-defined constraints. The Monitor is implemented using the Soar architecture as an agent architecture [11] and takes advantage of a highly efficient pattern matcher to evaluate the constraints against the learner-oriented description of the situation provided by the previous layers. These constraints were originally inspired by constraint-based expert modeling [12] but have been extended and customized for this function in the Dynamic Tailoring System. One specific example of a customization is a codification of distinct domain, scenario, and practice constraints [13], which enables (as one example) the monitor to assess the same learner action differently based on the specific goals of a practice exercise.

Figure 4 illustrates how the mapping and interpretation functions of the middleware components enable improved generality, ease of authoring, reusability, and transparency for the Monitor. Continuing the example from above, the Monitor's rules can leverage the general predicates `Angry()` and `InRegion()` to test for classes of events that should be reported, rather than needing to include simulator-specific tests for specific character grid locations and animations. The middleware lets the Monitor query simulation-specific inputs such as simulation state or physiological sensors and easily interpret the learner's behavior in order to determine not just whether the learner reacted correctly, but what underlying reasons might have caused any incorrect outcomes. The outputs of the Monitor (*shaded boxes*), which drive pedagogical decisions in the DTS, can be specified more abstractly allowing instructors to understand and control the system's behavior. Finally, the Monitor rules can be reused when new scenarios or new sensor input sources are added.

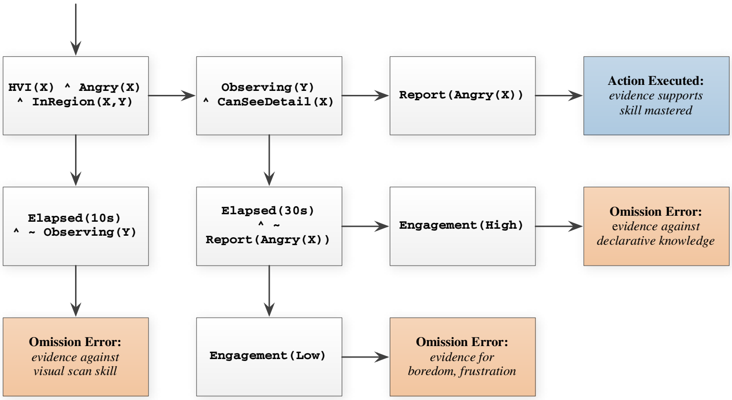

Fig. 4. Generalized predicates within the Monitor are able to describe a wide range of learner behaviors independently of simulation-specific details such as line-of-sight calculation and individual physiological sensor inputs

A significant benefit of the learner middleware approach is that the Monitor itself can be reused from one application to another without significant recoding. We have used the Monitor for applications as diverse as cross-cultural conversation, military decision-making and medical triage. A primary impetus for formalization of the middleware, as described in this paper, is our recognition that the constraint-based representation and pattern-matcher is proving powerful for many different applications.

Cognitive State Interpretation. The cognitive state interpretation function supports the Monitor but also provides direct measures of learner cognitive state and/or affect to other components of the DTS. Fig. 5 illustrates the cognitive state interpretation function. In this example, the dimension of interest is arousal/attention. The syntactic layer fuses sensor inputs and places individual observations on a normalized attention axis at a particular time, as outlined previously. The interpretation layer then compares the observations to a bounding "envelope" that defines the minimum and maximum desired levels for the dimension at a particular time.

The envelope provides a simple to use, actionable interpretation of cognitive state for other DTS components to use. An individual observation (or a prediction based on the trend/derivative) can help the DTS understand the relative priority and urgency of affective interventions. In the example in the figure, the learner's falling attention and the proximity of the current attention level to the lower bound of the envelope may lead the DTS to prioritize an attention-oriented tailoring strategy over a conceptual one. Similarly, the Pedagogical Manager might recommend an extrinsic intervention rather than in intrinsic adaptation in this situation because attention is sufficiently low that there is likely to be little interference with the learner's sense of presence in the practice experience.

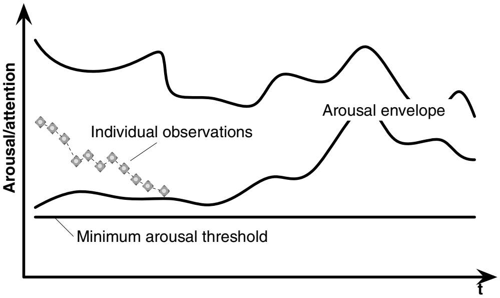

Fig. 5. Illustration of cognitive state interpretation

These envelopes today must be constructed by hand and adjustments made for different levels of proficiency (i.e., the envelope for the same task may be different for learners with different estimated levels of competency in the task). However, we are interested in methods that would allow us to construct them automatically and to compose envelope segments to accommodate learner actions and branching events within a scenario.

5 Conclusions

This paper has presented the conceptual design of general-purpose abstraction and translation layers to make it easier to obtain richer information from a practice environment than is typically afforded by a simulation environment. Although we noted several areas where the current implementations of this middleware are not yet fully developed or limited in their generality, the development thus far is providing benefit. We see two primary advantages to this learner-instrumentation middleware. First, it lowers the cost of integrating adaptive tailoring into a practice environment. Cost is reduced by supporting faster and simpler integration with simulation environments and by enabling reuse of the primary DTS components (Monitor, Pedagogical Manager, and Experience Manager) across applications. Second, it enables the integration of additional learner information streams, such as cognitive and affective state. The hypothesis is that these additional sources of information will enable more accurate diagnosis of the learner's needs and progress, extend the range of adaptation, and, ultimately, improve the efficiency of training.

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 project N00014-11-C-0193, Perceptual Training Systems and Tools (PercepTS).

References

1. Lane, H.C., Johnson, W.L.: Intelligent Tutoring and Pedagogical Experience Manipulation in Virtual Learning Environments. In: Cohn, J., Nicholson, D., Schmorrow, D. (eds.) The PSI Handbook of Virtual Environments for Training and Education, vol. 3, Praeger Security International, Westport (2008)
2. Schatz, S., Oakes, C., Folsom-Kovarik, J.T., Dolletski-Lazar, R.: ITS + SBT: A Review of Operational Situated Tutors. Military Psychology, special issue on current trends in adaptive training for military application (2012)
3. Schatz, S., Bowers, C.A., Nicholson, D.: Advanced situated tutors: Design, philosophy, and a review of existing systems. In: 53rd Annual Conference of the Human Factors and Ergonomics Society, pp. 1944–1948. Human Factors and Ergonomics Society, Santa (2009)
4. Tobias, S., Duffy, T.M. (eds.): Constructivist Instruction: Success or Failure? Routledge, Taylor and Francis, New York (2009)
5. Wray, R.E.: Tailoring Culturally-Situated Simulation for Perceptual Training. In: Duffy, V. (ed.) Proceedings of the 2012 Applied Human Factors and Ergonomics Conference, 2nd International Conference on Cross-Cultural Decision Making: Focus 2012. CRC Press, Taylor and Francis, Boca Raton, FL (2012)

6. Wray, R.E., Lane, H.C., Stensrud, B., Core, M., Hamel, L., Forbell, E.: Pedagogical experience manipulation for cultural learning. In: Workshop on Culturally-Aware Tutoring Systems at the AI in Education Conference, Brighton, England (2009)
7. Lane, H.C., Wray, R.E.: Individualized Cultural and Social Skills Learning with Virtual Humans. In: Durlach, P.J., Lesgold, A.M. (eds.) Adaptive Technologies for Training and Education, pp. 204–221. Cambridge University Press, New York (2012)
8. Schatz, S., Reitz, E., Nicholson, D., Fautua, D.: Expanding Combat Hunter: The science and metrics of Border Hunter. In: Interservice/Industry Training, Simulation and Education Conference (I/ITSEC), Orlando, FL (2010)
9. Gideons, C.D., Padilla, F.M., Lethin, C.R.: Combat Hunter: The training continues. Marine Corps Gazette, pp. 79–84 (2008)
10. Schatz, S., Wray, R., Folsom-Kovarik, J.T., Nicholson, D.: Adaptive Perceptual Training in a Virtual Environment. In: Human Factors and Ergonomic Systems (HFES 2012), Boston (2012)
11. Wray, R.E., Jones, R.M.: An Introduction to Soar as an Agent Architecture. In: Sun, R. (ed.) Cognition and Multi-agent Interaction: From Cognitive Modeling to Social Simulation, pp. 53–78. Cambridge University Press, Cambridge (2005)
12. Mitrovic, A., Ohlsson, S.: Evaluation of a constraint-based tutor for a database language. International Journal of Artificial Intelligence in Education 10, 238–250 (1999)
13. Wray, R.E., Woods, A., Priest, H.: Applying Gaming Principles to Support Evidence-based Instructional Design. In: 2012 Interservice/Industry Training, Simulation, and Education Conference, Orlando (2012)