Pattern Recognition for Cognitive Performance Modeling

Craig Haimson
Aptima, Inc.
1726 M Street NW
Suite 900
Washington, DC 20036
haimson@aptima.com

Stacy Lovell
Aptima, Inc.
12 Gill Street
Suite 1400
Woburn, MA 01801
slovell@aptima.com

Abstract

Cognitive performance modeling captures patterns of observable behavior that reflect the operation of underlying cognitive activities. These behavioral patterns can be linked to training requirements in order to derive automated measures of cognitive performance. We are currently developing the Team Coaching Assistant for Simulation-Based Training (T-CAST), a cognitive performance modeling system designed to support assessment of team performance in simulated environments. T-CAST maps competency-based performance requirements to data signatures of actions executed within a massively multiplayer online role-playing game. Through standard rule-based inferencing methods and spatial reasoning techniques, T-CAST produces lists of significant game-playing events from which aggregate measures of individual and team performance may be computed. These events are then fed back to the game for use in indexing simulation replay and supporting instructor-led after action review (AAR). In this paper we describe the design and development of T-CAST and suggest extensions of cognitive performance modeling to other domains and applications.

Introduction

The study of human cognition is largely the study of observable behavioral patterns that have been linked to unobservable internal processes. Cognitive mechanisms are by their very nature abstractions: explanatory constructs with certain demonstrable properties that can be inferred from patterns of overt motor responses or communications. Inferred mechanisms serve as theoretical explanations for observed behavior. These mechanisms can be formally modeled and experimentally validated as reliable predictors of behavioral patterns that emerge when human beings with specific abilities use a specific set of technologies to perform a specific set of tasks in a specific environment.

This material is based upon work funded by NAVAIR Orlando – TSD via Small Business Innovative Research Contract No. N61339-04-C-0021 under the management of the RDECOM STTC. Portions of this work were presented at the 15th Annual Conference on Behavioral Representation In Modeling and Simulation (BRIMS). We thank Mr. Brian Comer and Mr. James Grosse of PEO-STRI; Mr. Rodney Long, Ms. Michelle Mayo and Mr. Jeffrey Stahl of RDECOM STTC; and Mr. Bill Pike of the Joint ADL Co-Lab for their support of this effort. The opinions expressed here are the authors' and do not necessarily reflect the views of the United States Army or the Department of Defense.

Cognitive Performance Modeling

Cognitive performance modeling seeks to capture observable patterns of unobservable cognitive phenomena for use in monitoring and assessing human cognitive activity. It first leverages cognitive theory to postulate the mechanisms that underlie targeted cognitive skills. It then defines behavioral patterns that constitute the output of these cognitive mechanisms in specific operational contexts. If the link between a given pattern of behavior and an inferred skill is valid, then the ability to detect such "cognitive signatures" will have numerous applications for measuring and assessing human cognitive performance.

Behavioral patterns may be reliably coded by trained human observers, and, in many cases, subjective manual coding is the only method by which behavioral assessment is possible. However, many of the tasks performed by humans in the modern world utilize computer-based tools that enable the collection of data reflecting the operation of the system by the human. Such system usage data can provide an additional source of information from which cognitive indicators may be extracted. Cognitive performance modeling technologies can then be used to detect and assess these indicators automatically.

Simulation-Based Training

One of the most important applications of cognitive performance modeling is the evaluation of simulation-based training. Simulated worlds provide critical platforms for training and assessing human behavior. The military has made substantial investments in the development of synthetic task environments (STEs) that simulate operational tools, tasks, entities, and terrain for use during training and/or mission rehearsal. STEs can possess varying levels of physical realism, ranging from high fidelity flight simulators capable of a full range of motion, to low fidelity digital battlemaps that use patterns of icons to visualize the movement of forces within the simulated battlespace.

Whatever the level of physical fidelity, if a simulation evokes real-world decision making in a meaningful way, it can be said to possess adequate "cognitive fidelity" to provide useful training. While simulations cannot fully recreate the experience of being in an actual operational setting, they can construct situations that exercise the same skills of critical thinking, planning, communication, and coordination that are required for successful performance in the real world. Players demonstrate these skills by performing actions that change the status of their own simulated presence and/or that of other entities in the virtual world. These actions could entail the manipulation of physical controls analogous to those used in real-world interfaces (e.g., stick and pedals for a flight simulator), or they could involve the generation of mouse/keyboard commands that direct the behaviors of a player's avatar within the simulated environment.

The utility of a simulated training environment is tightly linked to its ability to support accurate and comprehensive performance assessment. Practice without feedback is unlikely to provide efficient or effective training experiences. Minimally, training should create opportunities to assess behaviors relevant to the operator competencies that the training is intended to address; ideally, such opportunities should be further tailored to address the specific learning requirements associated with individuals and teams participating in the exercise, with performance assessment and feedback that targets those learning objectives.

Military simulations are often conducted by experienced observer controllers (O/Cs) who monitor scenario progress, assess trainee performance, and facilitate the after action review (AAR) sessions at which these assessments are presented and discussed. Real-time observer-based performance measurement imposes a significant workload on the O/C, however. It is difficult to maintain a constant global awareness of all the activities that occur during a simulation; thus, evaluating trainee performance during simulation-based training exercises can pose a considerable challenge.

Luckily, virtual training platforms often maintain comprehensive records of the events that take place inside of their simulated environments. This makes it possible to use system data to support performance measurement and AAR. Raw lists of avatar actions executed during a scenario are unlikely to be very useful in this regard, as searching through a simulator data stream or log to identify important events for review can be an exceedingly difficult and time-consuming process. The high volume of data and the low level of granularity at which these data are presented make it difficult to identify critical incidents and to understand their implications through unaided human inspection alone. Thus, automated mechanisms are required to sift through logged data and synthesize them into a more useable form. Intelligent behavior recognition technology can provide this function, extracting higher level interpretations of simulated behaviors from clusters of lower level actions and events captured within a simulator data stream.

The T-CAST System

With our partners BBN Technologies and Forterra Systems, we are developing an automated cognitive performance modeling system for simulation-based training evaluation. We call this system the Team Coaching Assistant for Simulation-based Training (T-CAST). The US Army Research Development and Engineering Command Simulation Training Technology Center (RDECOM STTC) is supporting the development of T-CAST to provide automated AAR support for the Online Interactive Virtual Environment (OLIVE) massively multiplayer online role-playing game (MMORPG) (Figure 1), which Forterra Systems and RDECOM STTC are developing to train infantry teams for military operations in urbanized terrain (MOUT).

T-CAST is a reasoning system with a cognitive performance modeling capability. T-CAST is intended to be a *coaching* assistant – an automated tool that aids instructors and O/Cs in evaluating simulated team behaviors in order to support assessment and coaching during AAR. This is accomplished by operationalizing concepts of teamwork for evaluating avatar behavior it its spatial context.

Figure 1. Screenshot from the Online Interactive Virtual Environment (OLIVE) under development by Forterra Systems and RDECOM STTC.

Previous work in this area has addressed generating realistic human behavior simulations (Livak, Heffernan, & Moyer, 2004; Rickel & Johnson, 1999; Silverman, 2001; Wray & Laird, 2003) and inferring human behavior from simulated events (Sukthankar & Sycara, 2005). For example, in the MOUT domain, Best and Lebiere (2003) recently developed a cognitive architecture using ACT-R to represent and control planning, teamwork, and communication among synthetic agents. As in previous efforts, our approach utilizes behavior representation and event recognition models. These are augmented with representations of geo-spatial context, however, and mapped to training objectives and metrics.

T-CAST performs some of the functions of a classic intelligent tutoring system (ITS) (e.g., Sleeman & Brown, 1982), but the complexity of its student modeling is somewhat limited compared with ITSs designed to support fully automated training and feedback without any intervention from a human instructor. In sufficiently constrained contexts, such systems can map student actions to robust behavior models and generate highly refined diagnoses of underlying deficits in requisite knowledge and skills (e.g., Anderson & Corbett, 1993). However, in dynamic simulation-based training, task complexity prohibits the derivation of detailed cognitive models. Nevertheless, patterns of event data can be linked to carefully crafted measures of performance, which are, in turn, associated with specific competencies, knowledge, skills, and/or abilities. This is the approach that we have adopted for T-CAST.

To a great extent, the intended application of T-CAST – AAR support for observer/controllers – has shaped our modeling goals, as well. Our objective is not to create detailed models of behaviors but rather to create robust models of those events that serve as indicators of experiential learning. In some cases, patterns reflect discrete examples of correct or incorrect action; in others, patterns capture performance trends that can be evaluated through comparison with one of more standards (e.g., expected vs observed task completion time). In many cases, the ultimate interpretation of these events is left to the human O/C, who is better equipped to evaluate subtle context-specific nuisances. Thus, T-CAST is most properly thought of as a "behavioral consequence recognition" system or a "behavioral gist recognition" system.

The inclusion of a human in the behavior recognition loop reduces computational demands and provides some additional flexibility for dealing with unexpected and/or ambiguous situations. In some cases, T-CAST may be unable to determine the precise action that is taking place, or it may be unable to assess the correctness of the behavior; however, as long as it is capable of providing some indication that an attention-worthy activity has occurred at a certain simulated time/place, a human O/C will be able to inspect the action during AAR and render an informed judgment regarding its importance.

Approach

To develop T-CAST, we began by conducting a domain analysis to identify basic situations, tasks, and requirements for MOUT house-to-house search operations. We then constructed a competency-based event recognition and measurement model from these data and used it to code rule-based representations of critical events that can be inferred from OLIVE data logs. Finally, we developed methods to communicate between OLIVE's data store, T-CAST, and OLIVE's AAR replay module. We describe these steps in detail below.

Domain Analysis

The domain analysis consisted of evaluating many different resources to identify characteristics of effective MOUT behaviors. Specifically, we reviewed:

- Descriptions of competencies required of dismounted infantry teams in MOUT environments, as expressed within US Army field manuals (FM 3-06, FM 21-60, FM 21-75, FM 7-8), training doctrine, Lessons Learned reports, and interviews with subject matter experts (SMEs)
- Tactics, techniques, and procedures (TTPs) of cordon and search missions currently executed in Iraq. We attempted to generalize across different TTPs to uncover underlying commonalities.

We reviewed this material to identify key information for performance measurement: operational objectives, tasks that may be executed to meet those objectives, situations under which those tasks are executed (including characteristic non-combatant or hostile avatar behaviors), triggering conditions that bring on those situations, and the specific actions by which those tasks are carried out. We also identified common errors associated with the execution of such tasks and actions, as well as the potential consequences of such errors. We sought to characterize operational objectives, tasks and actions at the level of the individual soldier, fire team, and squad, and captured expected interactions and dependencies between the actions of different team members, including key communication patterns. Task and action data were organized in terms of primary soldier competencies that they leverage and the key knowledge, skills, and abilities (KSAs) with which they are associated.

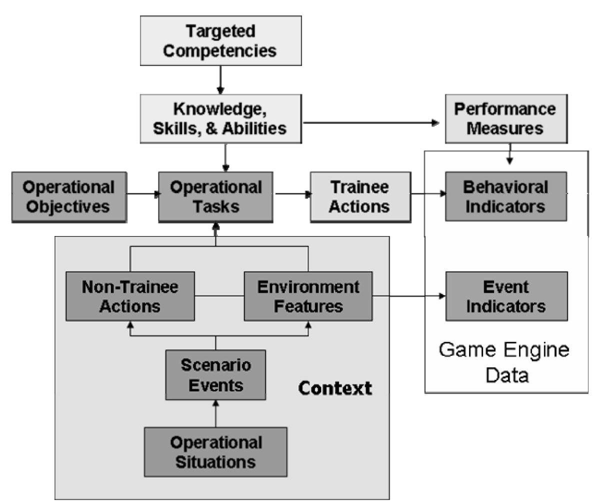

Figure 2. Competency-based event recognition and measurement framework.

We constructed a hierarchically-organized model of the various data collected in our analysis and devised a formal framework that captures the contents of this model (Figure 2). This model serves as a knowledge base containing situations, tasks, and actions that may be matched to events appearing within data collected by the simulator (e.g., Haimson, 2005).

We used this model to represent observable events that (1) are indicative of each situation, task, and action, and (2) could be demonstrated inside of MMORPGs (OLIVE). To facilitate this mapping, we developed storyboards for different situations and enacted associated tasks and actions within the OLIVE environment. We then mapped these actions to indicators of context (environmental features, non-trainee actions) and behaviors that could be made available within the OLIVE data log.

Cognitive Performance Modeling

The cognitive performance modeling technology underlying T-CAST utilizes artificial intelligence (AI) techniques that (1) recognize scenario events and avatar behaviors indicative of ongoing operational situations, (2) generate expectations for scenario events and behaviors indicative of the tasks and actions associated with recognized conditions, (3) compare attributes of expected and recognized tasks and actions, and (4) identify discrepancies between them in order to determine when noteworthy incidents have occurred. The knowledge base is populated with patterns of data associated with these events, and the event recognition technology utilizes sophisticated matching techniques to map these patterns to actual data collected by OLIVE during a scenario run.

The behavior/event recognizer operates as a forward chaining inference engine that iterates multiple times over a data set, each time deriving richer interpretations and higher level implications from inferences generated on previous cycles. Through this iterative re-description process, T-CAST is capable of identifying behavioral phenomena at different levels of granularity for individuals, fire teams, and squads. T-CAST can recognize that a given task has been accomplished by determining that characteristic sequences of actions were generated, as well as by recognizing that the objectives of tasks were achieved.

The recognition process (Figure 3) begins with the detection of simulator data patterns linked to specific low level events that are associated with operational situations. This prompts the recognizer to search for simulator data patterns linked to events associated with the tasks and actions that are expected to occur in response to these situations; note that a given set of situations can cue T-CAST for look for evidence of both correctly and incorrectly performed tasks and actions. T-CAST records the presence/absence of any and all expected tasks and actions (both correct and incorrect); if multiple interpretations of the data stream are generated, the event recognizer will search both earlier and later points within the data stream for further evidence to resolve the ambiguity. In many cases, T-CAST will be able to resolve ambiguous interpretations by examining data across the entire scenario. For cases in which T-CAST still lacks sufficient information to choose between multiple options, T-CAST will present all candidate interpretations as alternatives to be considered by the O/C. T-CAST does not currently include the capability to determine the best fitting rule given only partially matching data although this functionality is envisioned for the future.

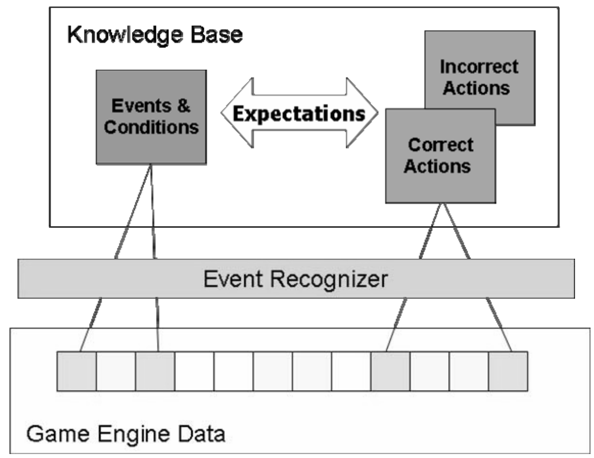

Figure 3. Event recognition process.

The T-CAST rules and evaluation functions are implemented in a combination of Jess (http://herzberg.ca.sandia.gov/) and Java. Patterns are represented as rule conditions that are matched to facts corresponding to raw data from the OLIVE engine and/or inferences derived through rule-based processing.

The T-CAST behavior/event recognizer is supported by additional modules that supply supplemental data for pattern analysis. Currently, we have implemented a spatial analysis module that leverages Java 3D models of the gaming environment to provide space-based interpretations of raw game data. The spatial analysis module develops a tagged three-dimensional map of the gaming environment from OLIVE collision mesh files and uses this map and information about avatar location and orientation to determine:

- Interpreted avatar locations in space
  - The names of tagged environmental locations in which avatars are positioned (e.g., "foyer")
- Line of sight/aim data
  - Objects within avatar viewpoint/gun aimpoint
- Team "stack" events

- Before entering uncleared rooms, fire teams line up in "stack" formation along the side of the wall containing the door that they are about to pass through
- Room clearance events
  - A room is defined as being cleared when a team of blue (own force) avatars has visualized (i.e., has had within its collective line of sight) the entire volume of the room within a certain tolerance threshold.
  - A room can revert back to uncleared status due to lack of timely/comprehensive coverage over time (e.g., if trainees' avatars fail to watch an exit/entrance to an uncleared room, the room in which they are currently standing may become uncleared due to danger of entry through the unguarded passageway)

The spatial analysis module runs prior to the startup of the main behavior/event recognizer. Many computations involving these spatial data are simple distance calculations that can be expressed and executed within the behavior/event recognizer; in the future, more involved computations may require additional external functions that are called by the main pattern matcher during runtime.

The following example demonstrates the first few events in a possible fire-team scenario with an objective of searching and clearing a building:

- First blue fire team stacks outside house and knocks on door
- Red (enemy force) inquires who it is
- Team leader requests permission to search
- Red denies request
- First fire team enters forcibly and clears room
- First fire team searches Red
- Second fire team enters…

These events have been encoded using expert system rules that capture avatar behavior and geo-spatial context (location of buildings, windows, doors, etc.). Assessing a team's performance in the above example would require various instances of behavior and event recognition, such as knowing:

- When a team is in a stack
- When blue avatars have cleared the "fatal funnel" (danger region in front of the door to an uncleared room)
- When a room is cleared
- When blue avatars are standing too close to a red avatar (exposes soldier to direct attack)
- When blue avatars are standing too close to a wall (exposes soldier to potential bullet ricochet)

These events must be inferred from low level data and previously recognized events. For example, knowing that the team has just cleared the fatal funnel requires correlating the positions of each avatar in the team with tagged locations in the spatial map, determining whether or not the room that is being entered is currently unclear, and determining when the last avatar in the team has passed the region designated as the funnel. The pattern matching algorithm will iteratively search the knowledge base for instances where the above conditions are met. It will then pass these events to evaluation functions in order to assess the validity of the behavior based on doctrinal baselines (e.g., if the team lead passed through the fatal funnel first, this would be counted as an error, as this violates prescribed order of entry).

Some of the events that T-CAST recognizes are merely treated as scene markers for parsing up action and indexing activities for replay. Other events serve as specific measures of individual behavior (e.g., specific errors that a single avatar generates). Still other events constitute aggregate measures of team performance that reflect group level behavior (e.g., spatial formation).

Our current T-CAST knowledge base encodes rules for recognizing events and computing measures focused on activities occurring during building search operations, including the following:

- Shot statistics
- Aimpoint
- Avatar health
- Room clearance (defined above)
- Coverage of avatars, objects, and key terrain
- Tactical configurations (e.g., stack)
- Tactical movement
- Distance/orientation from key landmarks and individuals
- Location within terrain (including position with respect to identified danger zones)

T-CAST also includes a set of aggregated performance functions that create descriptive summary statistics for certain classes of events (e.g., number of shots fired, number hits versus misses, etc.). As a default, these statistics are calculated across an entire scenario; however, it is possible to compute aggregations across any interval, including periods of time defined dynamically by the occurrence of specific events within the scenario. For example, it may be desirable to compute aggregations across a time window whose start and stop times correspond to:

- A particular scenario event that extends in time and has a duration that can be determined (e.g., period of time spent guarding a red avatar)
- The occurrence of two separate events (e.g., entering a room and completing a search)
- The occurrence of a particular scenario event and a set period of time preceding/succeeding this

event (e.g., one minute prior to a red avatar shooting a blue avatar).

Integration with OLIVE

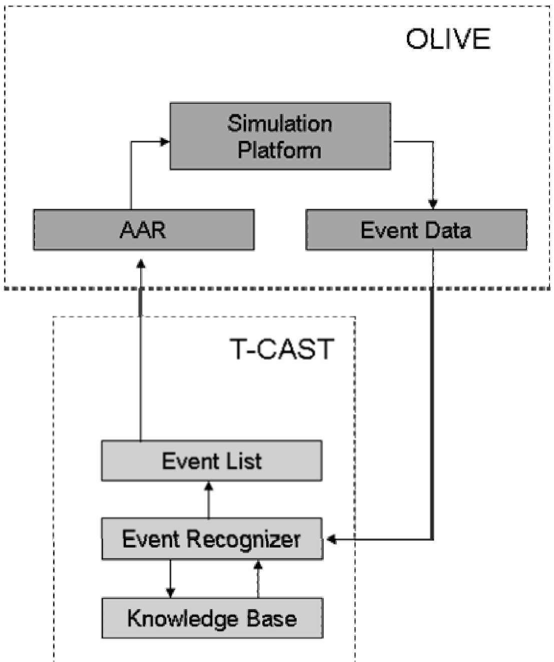

Figure 4. Interactions between T-CAST and OLIVE.

The primary interactions between OLIVE and T-CAST are shown in Figure 4. OLIVE's simulation platform generates low level event data and passes these data on to T-CAST's event recognizer. The event recognizer compares these data with the contents of its knowledge base and produces an event list (a time-stamped list of behaviors and events associated with situations, tasks, and actions that the event recognizer identified within the data stream). T-CAST passes the event list and aggregated performance statistics back as an XML file that can be displayed in conjunction with OLIVE's AAR module, which allows users to selectively replay segments of scenario footage.

Test and Evaluation

To test T-CAST's accuracy and reliability, we have developed a sequence of scripted scenarios that emphasize different behaviors represented within T-CAST's cognitive performance models. We have enacted these scenarios within OLIVE and examined T-CAST's outputs to verify that the behaviors, events, and assessments provided by T-CAST accurately reflect the expected behaviors of squad members in the scenario. We will be conducting a more complete assessment of T-CAST's performance in the Fall of 2006, in conjunction with OLIVE training exercises to be conducted with representative users from the operational community.

Conclusions

The T-CAST project is developing a cognitive performance modeling capability that facilitates teamwork training in simulated environments. T-CAST evaluates simulator data and flags data patterns indicative of critical situations, tasks, and actions that it identifies within them. We have been able to make progress on a number of activities thus far, including (1) analyzing MOUT search operations to craft models of critical scenario events to be recognized by T-CAST; and (2) developing formal representations for event models and implementing forward-chaining inference algorithms capable of identifying these patterns within the OLIVE event stream.

We intend for T-CAST to serve as general modeling framework for representing and detecting patterns of human responses that are indicative of cognitive performance. Human actions occur at specific times and in specific places. They form sequences that can uniquely indicate the status of the beliefs that the actor holds and the decisions s/he ultimately makes. Standard instructional analysis techniques can be used to define these behavioral patterns and to map them to validated psychological constructs; basic AI techniques can then be applied toward the recognition of these patterns in human performance data.

Simulated worlds for experiential learning provide particularly rich sources of data for cognitive performance modeling, supplying indicators of cognitively-significant behaviors and the context in which those behaviors occur. However, while we are developing T-CAST to support performance assessment during simulation-based training, we foresee many other applications for this technology. Any environment that provides adequate data from which patterns with cognitive validity can be extracted could serve as a domain in which cognitive performance could be applied. For example, cognitive performance modeling can increase the sophistication of user models that support automated software help features or facilitate marketing of web advertisements to internet users. In a distributed online work environment, cognitive performance modeling could serve as a means of recognizing user intent and facilitating coordination and communication among team members. Applied in reverse, the same technology could be used to interpret the actions of the users of an adversary's online command and control system. As more sources are incorporated into a dataset, new and more sophisticated technologies will be required to represent and detect meaningful data patterns. If, however, these patterns can be organized within a cognitive theoretical framework, their detection will provide a window into the decision making processes of the persons who generate them.

References

Best, B. J. & Lebiere, C. 2003. Spatial Plans, Communication, and Teamwork in Synthetic MOUT Agents. *In Proceedings of the 12th Conference on Behavior Representation In Modeling and Simulation.*

Field Manual (FM) 3-06.11, *Combined Arms Operations in Urban Terrain*, United States Army.

Field Manual (FM) 7-8, *Infantry Rifle Platoon and Squad*, United States Army.

Field Manual (FM) 21-60, *Visual Signals.* United States Army.

Field Manual (FM) 21-75, *Combat Skills of the Soldier.* United States Army.

Haimson, C. August, 2005. Competency-based training in gaming environments. Presented at the Joint Advanced Distributed Learning (JADL) Co-Laboratory's Implementation Fest 2005: Advances in Technologies and Learning, Orlando, FL.

Livak, T., Heffernan, N. T., Moyer, D. 2004. Using Cognitive Models for Computer Generated Forces and Human. Tutoring. *Proceedings of the 13th Annual Conference on (BRIMS) Behavior Representation in Modeling and Simulation*. Simulation Interoperability Standards Organization. Arlington, VA. Summer 2004.

Rickel, J. & Johnson, W. 1999. Virtual Humans for Team Training in Virtual Reality. In *Proceedings of the Ninth World Conference on AI in Education.* IOS Press.

Silverman, BG. 2001. *More Realistic Human Behavior Models for Agents in Virtual Worlds: Emotion*, *Stress, and Value Ontologies*, Philadelphia: U of Penn/ACASA Technical Report.

Sleeman, D., & Brown, SJ. (eds). 1982. *Intelligent Tutoring Systems.* Computers and People Series. Academic Press, Inc., London.

Sukthankar, G. & Sycara, K. 2005. Identifying Physical Team Behaviors from Spatial Relationships. In *Proceedings of 2005 Conference on Behavior Representation in Modeling and Simulation (BRIMS)*.

Wray, R.E., Laird, J. E. 2003. Variability in Human Behavior Modeling for Military Simulations, *Behavior Representation in Modeling and Simulation* Conference. Scottsdale, AZ.