Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

DEVELOPING AN AGENT MODEL OF HUMAN PERFORMANCE IN AIR TRAFFIC CONTROL OPERATIONS USING APEX COGNITIVE ARCHITECTURE

Seung Man Lee
Ujwala Ravinder
James C. Johnston

Human Factors Science and Technology Division
NASA Ames Research Center
Mail Stop 262-4
Moffett Field, CA 94035-1000, U.S.A.

ABSTRACT

For the analysis of large-scale complex systems, agent-based modeling and simulation has proven to provide a valuable research tool. The emphasis has, however, typically been on representing the dynamic behavior of physical entities such as aircraft. Simulation of human operators has often been minimal even though human behavior has an enormous impact on overall system performance and safety. Therefore, human capabilities and limitations need to be taken into account early in the system design process before irrevocable choices have been made. This paper reports on the development of agent models with human-like performance characteristics using a cognitive architecture. We present an agent model of an air traffic controller that is developed and incorporated into an agent-based simulation of the national airspace to support the design and evaluation of advanced air transportation concepts.

1 INTRODUCTION

Agent-based modeling and simulation (ABMS) is of increasing interest for the modeling and simulation of complex socio-technical systems such as the National Airspace System (NAS) (Lee 2002). A socio-technical system can be defined as a system consisting of a number of entities (such as humans, machines, technical systems, etc.) interacting with each other to accomplish their goals (Barrett et al. 2001). Examples of such systems include ground/air transportation systems, economic systems, manufacturing systems, health care systems, supply chain systems, and military systems. These systems are typically highly complex and dynamic with tightly coupled interacting components. The overall dynamic behavior of such a system typically emerges from the interactions among components. For example, the NAS is a large-scale, complex socio-technical system composed of controllers, pilots, airline dispatchers, aircraft, airports, navigation aids, and technical devices. Its behavior can be characterized by the individual dynamics of these different entities and their interactions. In the operations of the NAS, human controllers play an important role in maintaining safe and efficient flow of air traffic throughout the controlled airspace sectors (Wickens et al. 1997). Specific controller tasks include monitor ongoing flights, respond to various pilot requests, and adjust traffic in response to special conditions such as bad weather (typically by instructing pilots to alter their aircraft speed, flight levels, and headings) (Nolan 1999).

ABMS provides a natural methodology for representing and simulating the individual entities of complex systems (Wooldridge and Jennings 1995). The collective and emergent behavior of heterogeneous system components including hardware, software, and human operators can also be modeled and simulated as an interaction among model agents (Iglesias et al. 1999). While most research and development efforts have focused on accurately modeling the dynamics of physical components, less attention has been paid to the modeling and simulation of human behavior. However, humans are integral system components and critical elements affecting overall performance and safety of systems. Accurate modeling and simulation of large-scale complex systems require suitable agent models of human operators.

Recent developments in human performance modeling provide a new approach to understanding human cognition and behavior in computational terms at the level of whole systems, as well as the individual agent level (Lee et al. 2004). In ABMS, simulated human agents interact with computer-generated representations of the operating environment. More recently, researchers sought to incorporate simulated human agents into the design process to assess the design with respect to not only the physical attributes of human operator, but also the cognitive attributes (Eilbert et al. 1998). Human performance models allow designers to simulate human behaviors and responses in a variety of situations and with different design options. Such models can also be used to evaluate the impact of human behavior on system performance as well as the impact of changing technologies on the performance of human operators.

Computational modeling of human performance has significant value for human-computer interaction (HCI) and engineering applications. However, psychological models of human cognition and performance have typically been confined to simple HCI tasks, such as computer aided design, use of automated teller machines, and menu selection (Card, Moran, and Newell 1983). Some engineering models of human performance have been developed for large-scale agent-based simulations, but few of these have considered the cognitive limitations of human operators (Foyle et al. 2003).

The goal of the current efforts is to develop a computational model that can be incorporated into a complex, dynamic simulation environment. This paper outlines a computational approach to building agent models of human performance using the Apex (Architecture for Procedure EXecution) cognitive architecture developed at NASA Ames Research Center (Freed 1998). We describe how to construct a sequence of human behavior from elementary human behavior templates and how templates in our compositional approach might be structured to accommodate underlying human behavior. Our task analysis of human performance observed in a simulation of air traffic control operations conducted by the Federal Aviation Administration (FAA) is described. We also present an example of an agent model of an air traffic controller that is developed and incorporated into an agent-based simulation of the national airspace to support the design and evaluation of advanced air transportation concepts.

2 AGENT-BASED MODELING AND SIMULATION OF HUMAN PERFORMANCE

As systems become more complex, it becomes harder to anticipate all the potential interactions that can occur from a priori analysis. Recent developments in software engineering, artificial intelligence, human-machine systems, and simulation science have placed an increasing emphasis on the concept of agent-based modeling and simulation (Davis, Sloman, and Poli 1995).

In ABMS, agent models can be heterogeneous, i.e., a variety of agent models may be included in a simulation to represent the diverse types of entities and their behavior. Agents can be physical entities such as people, animals, vehicles, and machines in the real world, or agents can be task-oriented entities such as strategic planning, scheduling, monitoring, communications, and decision-making activities. Agents used in agent-based simulation can also include rich cognitive human models and sophisticated communication and interaction mechanisms.

Agents can be used to capture the accuracy, speed, and variability of human performance, which are critical to the safety and performance analysis of the larger system. A computational agent model of human performance can be defined as a representation of human behavioral characteristics that can be implemented and executed in a simulation environment.

Modern technology has increased the importance of cognition in the design of complex systems. Thus, human performance models have increasingly incorporated the perceptual, cognitive, and motor capabilities and limitations of human operators. Although theories of cognition are presently incomplete and tentative, several recent studies have explored the development of agent models of human performance for analysis of large-scale complex systems (Pritchett et al. 2002, Jones et al. 1999, Callantine 2001). These human agent models drive high-level behaviors by applying domain knowledge. However, they do not attempt to model the cognitive capabilities and constraints on human performance.

Humans can also be considered as a system of systems consisting of perceptual, cognitive, and motor systems, with the behavior of each human itself emerging from the interaction between the subsystems (Barnard 1985). Modules dedicated to perception, attention, working memory, and decision-making interact with each other to generate human behavior. Each module of human behavior is limited in its ability to process information (Card et al. 1983). Agent models of human operators need to encompass psychological findings and theories to generate human-like behavior in psychologically plausible ways. Though much is known about the limitations of a single sub-system, the interaction among sub-systems makes it difficult to predict how people will respond in complex task environments such as an air traffic control (ATC) system.

The level of detail to which each human agent needs to be modeled depends upon the purpose of the simulation model. A simulation that is too detailed costs more and may only complicate the evaluation procedure, whereas a simulation that is too shallow provides insufficient or misleading information. Given the complexity of human agent models, creating the ability to interact with other simulation models (e.g., communicate with other agents and synchronize their time advance with the other agents) can require significant adaptations. Ultimately, it is hoped that these simulations will have sufficient fidelity in their agents' ability to reason and react to unexpected situations to examine a wide range of potentially hazardous situations. It should also be noted, however, that even examining the agents' behavior in normal circumstances can identify potential weakness or inconsistencies in standard operating procedures.

There are several critical issues in developing agent models of human behavior and cognition. First, models must be able to generate human-like behavior in psychologically plausible ways. Agent models provide insight into the impact of human performance on overall system performance and predict global consequences of system changes. Second, models must be easily modifiable and quickly adaptable to new situations. Existing human performance models typically require deep knowledge of human behavior and cognition and also require a great amount of time and effort to develop a new human performance model for a specific domain of interest. Third, models should be implemented in a computational architecture with the capability for representing the range of behavior found in complex task domains, and that can be incorporated into a large-scale simulation environment to interact with other agents and can be run in a real-time fashion (Bass 1995).

3 DEVELOPING HUMAN AGENT MODELS

To build a computational agent model of human performance in complex domains it is essential to analyze the task structure at the level of resource utilization. Structured task analysis provides a detailed knowledge of the tasks to be performed, an understanding of the current system, and information flow within the system. The task analysis approach also enables an appropriate allocation of tasks and functions to be included within new systems. Our task analysis consists of a hierarchical task decomposition based on the Goal, Operators, Methods, and Selection (GOMS) technique (John 1990).

CPM-GOMS (Cognitive, Perceptual, Motor -GOMS) combines GOMS with a cognitive architecture called the model human processor (MHP). MHP makes an assumption that human behavior can be described as the interaction of cognitive, perceptual and motor systems. The CPM-GOMS task analysis consists of a set of goals that the human is trying to achieve. The top-level goal is decomposed into subgoals, and these are decomposed into low-level CPM operators. Operators describe the perceptual, cognitive and motor actions that are required to accomplish the task. Methods describe the procedural knowledge required to complete the tasks. When several methods compete for task completion, selection rules predict which method the human operator would select in a given situation.

3.1 Template-Based Modeling Approach

In general, detailed models of human performance are difficult and time-consuming to build and require specialized knowledge about human cognition and behavior. To facilitate model construction, our approach is to decompose a complex task into a set of primitive task-level operations and develop common, reusable building blocks, *templates*, that model fundamental human cognitive, perceptual and motor behaviors that recur in multiple task domains (Matessa et al. 2002). Representing human behaviors in terms of the fundamental-level operators allows very accurate prediction of human performance.

Templates are psychological models of elementary human cognitive, perceptual, and motor behaviors, (e.g., monitoring a screen and detecting an event, typing, moving a mouse and clicking a button, etc.), that are common across task domains. By integrating a theory of composition from templates, larger models of human performance can be created without requiring modelers to have a deep understanding of cognitive psychology. A standard library of domain-independent templates can be applied in different circumstances. Templates eliminate the need for developers to have extensive knowledge of the underlying cognitive architecture. Consequently, the use of templates makes cognitive modeling more accessible to a wider range of domain experts.

There are two issues that add to the complexity and difficulty of the compositional approach: template construction and composition of extended behavioral sequences from templates. In constructing reusable and scaleable templates, the choice of primitives and the method of combining basic cognitive, perceptual, and motor operations into larger behavior units are critical. To construct templates for modeling human performance, it is necessary to identify those portions of the behavioral stream that contain routine behavior through a detailed task analysis.

Apex composes extended behavioral sequences by automatically interleaving templates for successive behaviors. We have recently described an approach for automatically generating the sequence of behavior using the Apex cognitive architecture (John et al. 2002). The automation in Apex makes it possible to derive detailed predictions of human performance with complex tasks and interfaces.

3.2 Apex Cognitive Architecture

Apex is designed for generating adaptive, intelligent human behavior in complex, dynamic environments. Apex incorporates many high-level aspects of cognition including action selection under uncertainty, managing multitasking, and task interleaving, which allows modeling of a wide range of human behaviors common to complex domains. Apex models a human operator engaged in multiple tasks, and decides how to allocate limited resources to accomplish these tasks. The high-level architecture of Apex is shown in Figure 1.

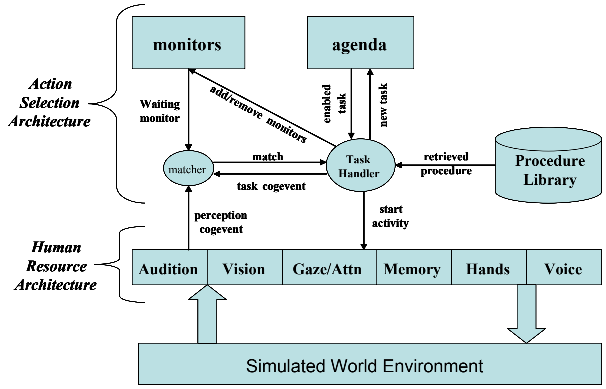

Figure 1. Apex Agent Architecture

The Apex framework includes a *Human Resource Architecture*, an *Action Selection Architecture*, and a *Procedure Library*. The *Human Resource Architecture* defines limited-capacity cognitive, perceptual, and motor components. The *Action Selection Architecture* coordinates the use of the resources, enforces constraints on resource allocation, and applies domain knowledge. It determines which tasks should be active and how resources should be allocated. Tasks become active when events match the conditions on a procedure in Apex's Procedure Library**.** The *Procedure Library* contains domain knowledge and a set of tasks to be performed in the target domain. The knowledge is represented in the form of procedures.

4 A MODEL OF AN EN-ROUTE AIR TRAFFIC CONTROLLER

Building a human performance model for a complex domain such as air traffic control requires in-depth understanding of the task structure in that domain. In order to develop a model of an air traffic controller responsible for traffic in a single en-route sector, we examined the tasks and procedures performed by controllers based on existing functional task analyses (Seamster et al. 1993, Leiden 2002, Niessen, Leuchter, and Eyferth 1998).

In routine operations, a radar en-route controller monitors the flights passing through the sector, responds to various pilot requests, and adjusts aircraft trajectories by instructing pilots to maintain the safe and efficient flow of air traffic within his sector. A high level functional analysis of current en-route control operations is presented in previous work (Remington et al. 2004). The most important high level tasks of the controllers can be classified into accepting handoffs of the aircraft from other sector, initiating transfer of control of the aircraft to the other sectors, monitoring aircraft, detecting and avoiding conflicts, detecting and resolving metering violations, attending to pilot requests and communicating with the pilots and other neighboring controllers. Typically, many tasks of controllers can be performed in parallel. For example, controllers perform a handoff task while concurrently giving a clearance command to a pilot and monitoring radar screen for other events.

With little effort a hierarchical task analysis can be represented as a GOMS task analysis. In our template-based modeling approach as shown in Figure 2, the high-level tasks are recursively decomposed into the template-level of elementary behaviors such as move and click behavior, scrolling, speaking, and typing and so on. The low-level templates are typically domain independent. Our approach is to combine and interleave low-level templates to build models of human performance in any domain.

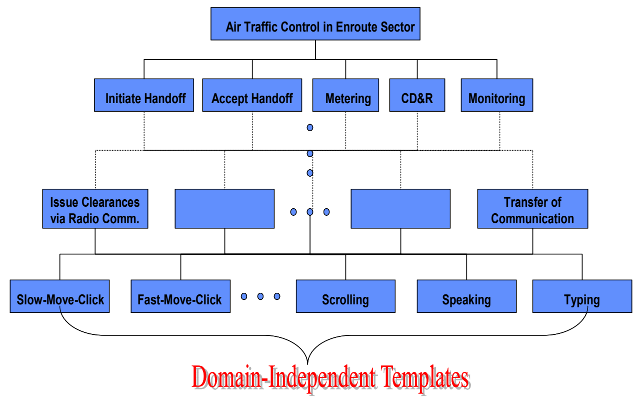

Figure 2. Hierarchical Task Decomposition

Based on the task analysis, we developed a simple Apex model of an en-route sector controller for a handoff task. Figure 3 shows a detailed task analysis of accepting a handoff. We have constructed templates (e.g., move-and-click, typing, and speech templates) for handoff operations based on the task analysis. The model predicts time and resource usage, both of which are necessary to provide insight into the mental demands placed on the controller in routine operations. This model permits estimates of workload, throughput, and suggest efficient ways to structure tasks. In our Apex model, a set of resource constraints is implemented to mirror human performance limitations. Tasks that require the same resources can temporarily prevent parallel execution of multiple tasks. For example, a controller can visually detect two handoffs simultaneously, but can only accept one handoff at a time because the task requires a limited resource, the use of a trackball. This illustrates how the resources in our Apex model capture constraints on human behavior, which have consequences on overall performance.

Description at the level of resource utilization adds value by providing insight into issues such as workload and throughput, hence we wish to retain this essential character of CPM-GOMS by first identifying and modeling those portions of the behavioral stream that contain routine behavior from detailed task analysis. Model parameters like durations of operations are estimated from existing theories and analyses, which represent zero-parameter predictions of performance. For example the mouse-move-and-click template includes finding the target, fixating on the target, moving the mouse to the target and clicking the target. In this template Fitt's law is used to compute the time taken to move the mouse to the target.

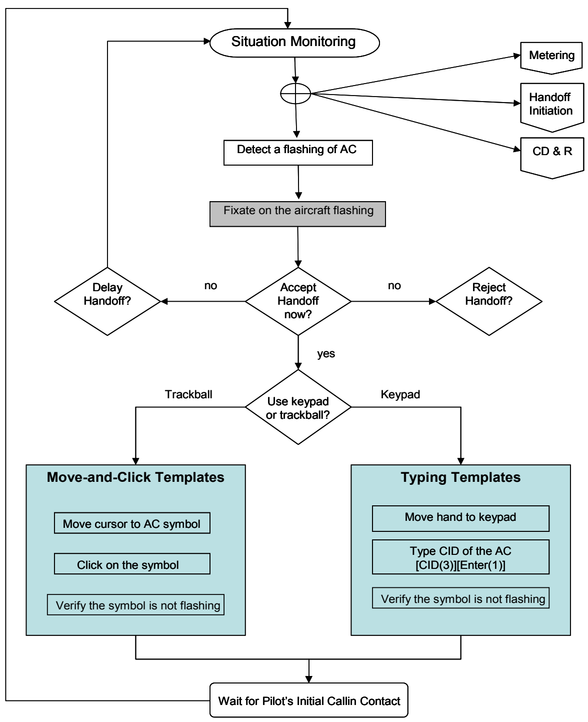

Figure 3. Task Analysis of Accepting a Handoff

From the reusable templates we can start building templates that are domain-specific. For example, typically when the aircraft icon on the radar screen starts flashing, this indicates the controller that the aircraft is ready to be accepted into their sector. The template of accepting aircraft would include fixating on the flashing aircraft, deciding to accept the aircraft, moving the mouse and clicking on the aircraft icon. This detailed task analysis can be represented in Apex as a domain specific procedures for accepting handoff which bottoms out to domain independent templates.

The hierarchical goal structure of a GOMS model is expressed in Apex using its Procedure Description Language (PDL). PDL steps are decomposed hierarchically into procedures of simpler steps until those steps bottom out in primitive actions that occupy human resources. Figure 4 shows high-level PDL procedures for the handoff task that decompose into low-level procedures. Figure 5 illustrates a template-level procedure written in PDL. This level of procedure represents very detailed human cognitive, perceptual, and motor behaviors. This move-and-click template was an existing template developed for a different task. This existing template was reused to develop a human performance model for handoff task.

```
(procedure
 (index (detect initiating handoff ?aircraft))
 (step s1 (decide whether to initiate handoff ?aircraft))
 (step s2 (initiate handoff ?aircraft to next controller)
          (waitfor ?s1))
 (step s3 (monitor response from receiving controller)
          (waitfor ?s2))
 (step s4 (issue frequency change to pilot) (waitfor ?s3))
 (step s5 (mark ac shipped)
          (waitfor ?s4 (pilot readback)))
 (step done  (terminate) (waitfor ?s5)))

(procedure
 (index (receive handoff request for ?ac-symbol))
 (step s1 (acquire sa for handoff ac))
 (step s2 (determine response) (waitfor ?s1))
 (step s3 (respond to initiating controller ?ac-symbol)
          (waitfor ?s2))
 (step s4 (wait for initial contact from pilot)
          (waitfor ?s3))
 (step done (terminate) (waitfor ?s4)))
```

Figure 4. High-level Apex Task Procedures

```
(procedure
 (index (slow-move-click ?target))
 (step c1  (initiate-move-cursor ?target))
 (step m1 (move-cursor ?target)                    (waitfor ?c1))
 (step c2  (attend-target ?target))
 (step c3  (initiate-eye-movement ?target)         (waitfor ?c2))
 (step m2 (eye-movement ?target)                   (waitfor ?c3))
 (step p1  (perceive-target-complex ?target)       (waitfor ?m2))
 (step c4  (verify-target-position ?target)        (waitfor ?c3 ?p1))
 (step c5  (attend-cursor-at-target ?target)       (waitfor ?c4))
 (step w1  (WORLD new-cursor-location ?target)     (waitfor ?m1))
 (step p2  (perceive-cursor-at-target ?target)     (waitfor ?p1 ?c5 ?w1))
 (step c6  (verify-cursor-at-target ?target)       (waitfor ?c5 ?p2))
 (step c7  (initiate-click ?target)                (waitfor ?c6 ?m1))
 (step m3 (mouse-down ?target)                     (waitfor ?m1 ?c7))
 (step m4 (mouse-up ?target)                       (waitfor ?m3))
 (step t1  (terminate)                             (waitfor ?m4)))
```

Figure 5. Template-level Apex Task Procedure

5 INTEGRATED APEX-ACES SIMULATION

In the Virtual Airspace Modeling and Simulation (VAMS) project, a joint NASA and FAA effort, modeling and simulation methods are being applied to evaluate the effect of changes in the operation of the national airspace. To this end, the Apex agent architecture has been integrated with the ACES (Advanced Concepts Evaluation System) airspace simulation system

ACES is a fast-time distributed agent-based simulation for the analysis of the NAS. The ACES simulation emulates an entire day of operations in the NAS with agent models for aircraft, Air Traffic Control System Command Center (ATCSCC), ARTCC, Traffic Flow Management (TFM) & Air Traffic Control (ATC), Terminal TFM & ATC, Airport TFM & ATC, traffic demand, and weather.

These different types of agents interact with each other through message passing during the simulation. However, ACES does not have agent models of human operators, such as air traffic controllers and pilots.

In order to allow Apex models to participate in large-scale simulations of the national airspace system, our approach was to integrate the agent model of an en-route air traffic controller developed using the Apex architecture into the ACES simulation environment (Lee et al. 2004). Figure 6 shows a high-level framework of the integrated Apex-ACES system for air traffic control simulations. Conceptually, ACES provides a simulation environment including agent models for the operations of the NAS. Apex provides a model of an air traffic controller for receiving and interpreting information coming from the ACES simulation world and for taking appropriate actions, as a human controller does.

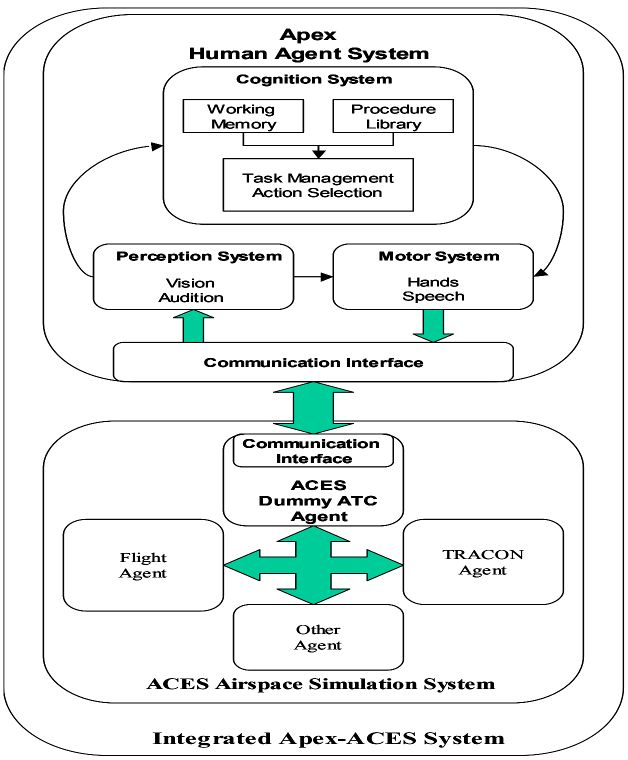

Figure 6. A Schematic of Integrated Apex-ACES System

To enable communication between these two systems we developed communication interfaces and message protocols using TCP/IP sockets. As shown in Figure 6 the Apex human agent model interacts with other agents in the ACES simulation environment through a *Communication Interface*. Information about the external world and events are received through the communication interface and stored into a limited-capacity working memory through the perception system of the Apex agent. Incoming information is matched against the specifications of procedures in the procedure library of the Apex agent. If the conditions for a procedure are met then the action selection architecture of the Apex agent schedules the steps of the procedure in accordance with the constraints and task management mechanisms. To control the aircraft's motion the Apex agent issues commands to a corresponding ACES agent. The interface converts the commands to a message format, which can then be interpreted by ACES agents. Currently, an aircraft is controlled through setting the desired state of the aircraft (i.e., speed, heading, and altitude).

Apex is used to model both the human agent and the displays and controls the agent interacts with. In this way, timing of agent actions and delays imposed by the equipment and user interface can be simulated at high fidelity. Communication with the ACES simulation will occur at synchronized message passing times in accordance with ACES protocols. This scheme provides a natural division between the human performance model and the ACES-level agent. The Apex agent only needs to transmit to its ACES counterpart those messages of significance in the larger context. Thus, while each keystroke of data entry into flight computers must be simulated to predict the time, the ACES-level agent need only be informed when the keystrokes produce some change in the its state, such as activating a mode, or changing a control setting.

Figure 7 shows a snapshot of the integrated Apex–ACES simulation. For this simulation, an agent model of an en-route air traffic controller was developed using Apex from task analyses and other literatures for high-level routine controller tasks (i.e. accepting handoffs, initiating handoffs, resolving metering violation, monitoring, the progress of aircraft, and conflict detection and resolution tasks) (Remington et al. 2004). The agent model of an air traffic controller developed was successfully integrated into the ACES agent-based simulation to interact with other agents and accomplished tasks with a simple traffic scenario.

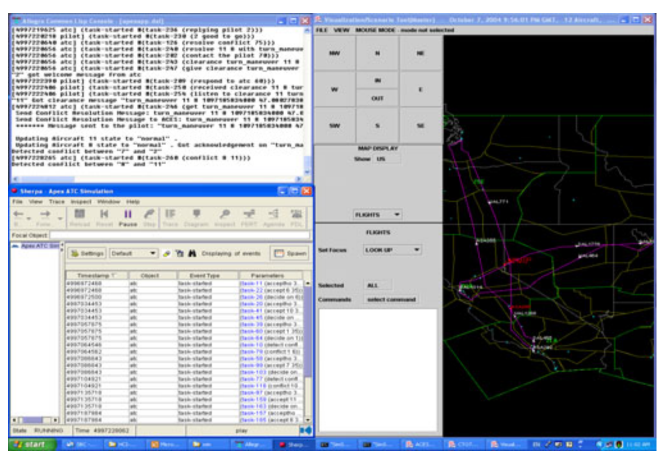

Figure 7. A Snapshot of Integrated Apex-Aces Simulation

6 VALIDATION OF APEX TEMPLATE MODELS

Templates in the Apex enroute controller model have been validated with the human-in-the-loop simulation data from the FAA. Comparison of the results show that the agent's performance fell in the range of best performance of highly skilled controllers.

We compared the predicted times of Apex model for accepting handoff with the human-in-the-loop simulation data from FAA, as shown in Figure 8. The task time for accepting handoff means the time from first fixating the aircraft to the acceptance of the handoff. After fixating on the aircraft, the controller model accepts it by moving a trackball cursor over the target and clicking the button. The times for these actions were taken from existing theory and data (Leiden 2000). They are fixed in the model, not estimated from the observed simulation times, providing zero-parameter predictions. On average, the model predicts the total task time of accepting the handoff accurately and provides at least the fastest task response time of experienced air traffic controllers.

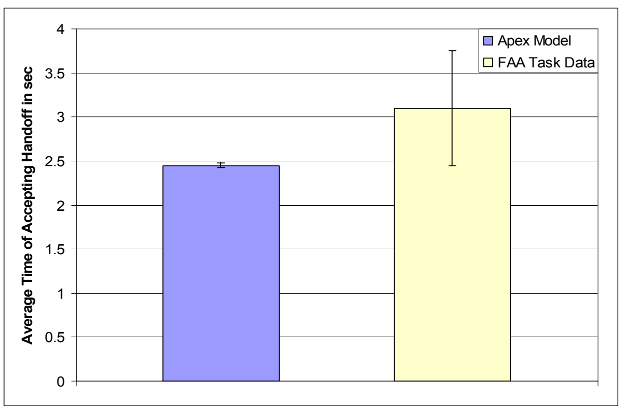

Figure 8. Average Task Time of Accepting Handoff

Figure 9 plots observed simulation data and predicted times of the communication duration between air traffic controllers and pilots. The communications include initial contact, welcome acknowledgment, and speed, altitude, and heading change commands. A speaking template developed in this study predicts the total speaking (utterance) time based on the number of syllables where it is assumed that each syllable takes 200ms. In most cases, as shown in Figure 9, the template predicts speaking times well. Empirical simulation data show slightly longer times of utterance than the predicted times of the model. The correlation of the simulation data and predicted time was 0.837.

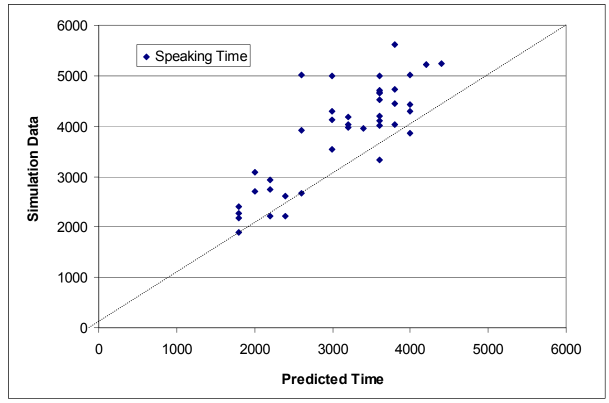

Figure 9. Comparison of Speaking Times

In both the cases the reason for the empirical data to show longer times than that predicted by the model is that the templates predict the fastest task completion time. Another reason might be the controller was engaged in a higher priority task such as conflict resolution, metering task, or next tasks to be performed immediately while talking to a pilot. For example, we noticed that while communicating vocally with a pilot, controllers sequentially fixated data blocks or executed a handoff acceptance from other aircraft. However, it was hard to capture those cases where the controller suspended an on-going task (interrupted by a higher priority task) and resumed the interrupted task when the higher priority task was completed. These high-level multitasking behaviors require further investigation

7 CONCLUSIONS

Computational agent models of human performance hold the potential to provide a safe and cost-effective way to test the design and implementation of new technology, predict possible human errors, as well as to anticipate transitional challenges and their impact on implementing new task procedures.

This article described a template-based approach for developing agent models of human performance in complex domains. We have shown how computational agent models of human performance can be built with human performance templates from a GOMS task analysis in the Apex cognitive architecture. We have also validated the templates with data from the FAA Tech Center. Our results show that, without estimating parameters from the task, our Apex model was able to make accurate estimates of the time taken to execute the acceptance of a handoff and the time of utterances. The model developed largely from task analysis is now being expanded and modified in response to patterns observed in the simulation data.

It was also demonstrated that human performance models using the Apex architecture can be extended to interact with a larger agent-based simulation environment. The Agent-based modeling and simulation approach may provide an efficient and effective way of developing and simulating human performance models. However, the decision of what should be modeled as an agent in ABMS is not always clear-cut. As described, human entities within the system under investigation can be modeled as agents, or agents can be defined around functional attributes and tasks (Bonabeau 2002). For example, each agent may represent the behavior of one human in the system or different agents may handle different tasks involving multiple humans, such as having one agent handling a negotiation activity, another handling communications, etc. Therefore, the trade-off between the level of decomposition and efficiency in terms of the design and modeling of individual agents should be further investigated. The interactions between agents should be carefully considered in designing agent models.

In order to develop higher fidelity human performance models, we must have detailed human performance data, greater psychological understanding of fundamental human behavior, and improved architectures for building human performance models efficiently. Our future goals include expanding our template library to cover more domain-independent activities, and to induce more proactive behavior by enhancing the model's monitoring, spatial reasoning capabilities, and dynamic multitasking behaviors. It is expected that the template library will allow model developers develop agent models of human performance in a cost and time efficient way.

ACKNOWLEDGMENTS

This work was supported by the Virtual Airspace Modeling and Simulation project of the NASA Airspace Systems program. The authors wish to thank Lisa Bjarke, Karlin Roth, and Larry Meyn for programmatic assistance, and Parimal Kopardekar, Ben Willems and Ken Leiden for making available task analysis and task modeling data.

REFERENCES

Barnard, P. 1985. Interacting cognitive subsystems: A psycholinguistic approach to short term memory. In *Progress in the Psychology of Languages*, A. Ellis, Ed. Hove, United Kingdom: Lawrence Erlbaum Associates. 197-258.

Barrett, C. L., S. G. Eubank, M. V. Marathe, H. S. Mortveit, and C. M. Reidys. 2001. Science & engineering of large scale socio-technical simulation. Los Alamos National Laboratory Report LA-UR-01-6623.

Bass, E. J., G. D. Baxter, and F. E. Ritter. 1995. Creating models to control simulations: A generic approach. *AI and Simulation of Behaviour Quarterly* Vol. 93:18-25.

Bonabeau, E. 2002. Agent-based modeling: methods and techniques for simulating human systems. In *Proceedings of National Academy Science (PNAS)* 99 (3): 7280-7287.

Callantine, T. 2001. Agents for analysis and design of complex systems. In *Proceedings of the 2001 International Conference on Systems, Man, and Cybernetics*, 567-573.

Card, S. K., T. P. Moran, and A. Newell. 1983. *The psychology of human-computer interaction*. Hillsdale, N.J.: L. Erlbaum Associates.

Davis, D. N., A. Sloman, and R. Poli. 1995. Simulating agents and their environments. *AISB Quarterly*.

Eilbert, J. L., G. E. Campbell, T. Santoro, T. L. Amerson, and J. A. Cannon-Bowers. 1998. The role of cognitive agents in the design of complex systems. In *the 1998 Annual Interservice/Industry Training Systems and Education Conference*, Orlando, FL.

Foyle, D. C., A. Goodman, and B.L. Hooey. 2003. Proceedings of the 2003 NASA Aviation Safety Program Conference on Human Performance Modeling of Approach and Landing with Augmented Displays, NASA Ames Research Center, Moffett Field, California; March 6.

Freed, M. 1998. *Simulating human performance in complex, dynamic environments*. Doctoral dissertation, Northwestern University.

Iglesias, C. A., M. Garijo, and J. C. Gonzalez. 1999. A survey of agent-oriented methodologies. in *Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages*.

John, B. E. 1990. Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. In *Proceedings of CHI*, Seattle, Washington, 107-115.

John, B. E., Vera, A. H., M. Matessa, M. Freed, and R. Remington. 2002. Automating CPM-GOMS. in *Proceedings of CHI'02: Conference on Human Factors in Computing Systems*: New York, ACM Press, 147-154.

Jones, R. M., J. E. Laird, P. E. Nielsen, K. J. Coulter, P. Kenny, and F. V. Koss. 1999. Automated intelligent pilots for combat flight simulation. in *AI magazine*, 27-42.

Lee, S. M. 2002. *Agent-based simulation of socio-technical systems: Software architecture and timing mechanisms*. Doctoral dissertation, Georgia Institute of Technology.

Lee, S. M., R.W. Remington, U. Ravinder, & M. Matessa. 2004. Developing human performance models using Apex/CPM-GOMS for agent-based modeling and simulation. In *Proceedings of the 2004 Advanced*

Simulation Technologies Conference (ASTC'04), Arlington, VA.

Leiden, K. 2000. Human Performance Modeling of En Route Controllers. Micro Analysis & Design, Inc., Boulder, CO., RTO-55 Final Report, Prepared for NASA Ames Research Center, December.

Matessa, M., A. Vera, B. E. John, R. Remington, M. Freed. 2002. Reusable templates in human performance modeling. In *Proceedings of the Twenty-fourth Annual Conference of the Cognitive Science Society*.

Niessen, C., S. Leuchter, and K. Eyferth. 1998. A psychological model of air traffic control and its implementation. in F. E. Ritter and R. M. Yong (eds), in *Proceedings of the Second European Conference on Cognitive Modeling*, Nottingham, U.K., 104-111.

Nolan, M. S. 1999. *Fundamentals of air traffic control*, 2nd ed. Belmont, Calif.: Wadsworth Pub. Co..

Pritchett, A. R., S. M. Lee, K. M. Corker, M. A. Abkin, T. G. Reynolds, G. Gosling, and A. Z. Gilgur. 2002. Examining air transportation safety issues through agent-based simulation incorporating human performance models. in *Proceedings of the IEEE/AIAA 21st Digital Avionics Systems Conference*.

Remington, R.W., S. M. Lee, U. Ravinder, & M. Matessa,. 2004. Observations on human performance in air traffic control operations : Preliminaries to a cognitive model. in *Proceedings of the 2004 Behavioral Representation in Modeling and Simulation (BRIMS'04)*, Arlington, VA.

Seamster, T. L., R. E. Redding, J. R. Cannon, J. M. Ryder; J. A. Purcell. 1993. Cognitive task analysis of expertise in air traffic control. *International Journal of Aviation Psychology* Vol. 3: 257-283.

Wickens, C. D., A. S. Mavor, J. McGee. 1997. National Research Council (U.S.), Panel on Human Factors in Air Traffic Control Automation, *Flight to the future : human factors in air traffic control*. Washington, D.C.: National Academy Press.

Wooldridge, M. J. and N. R. Jennings. 1995. Intelligent agents: Theory and practice. in *Knowledge Engineering Review* 10 (2): 115-152.

AUTHOR BIOGRAPHIES

SEUNG MAN LEE is a senior project associate conducting cognitive modeling and simulation research at NASA Ames Research Center. He earned his Ph.D. in Industrial and Systems Engineering from Georgia Institute of Technology. His research interest includes the modeling and simulation of human cognition and behavior, and his work has emphasized the application of agent models of human performance to aerospace domains. His email address is smlee@mail.arc.nasa.gov.

UJWALA RAVINDER received her M.S. degree in Computer Science from University of Minnesota, Twin Cities. She is working as a senior research associate in the cognition laboratory at the NASA Ames Research Center. Her current work is focused on task analysis of complex domains, developing and analyzing human performance models in the aviation domain. Her email address is uravinder@mail.arc.nasa.gov.

JAMES C. JOHNSTON is the Group Lead for Cognition Laboratory at the NASA Ames Research Center. He received his Ph.D. in Human Experimental Psychology from the University of Pennsylvania in 1974 He worked for Bell Laboratories in Murray Hill, N.J before joining NASA in 1985. His research focuses on human visual attention and multitasking. He has worked to integrate cognitive psychology research into computational models of human performance, especially applied to aerospace domains. His email address is jcjohnston@mail.arc.nasa.gov.