AGENTS FOR ANALYSIS AND DESIGN OF COMPLEX SYSTEMS

TODD J. CALLANTINE

San Jose State University/NASA Ames Research Center
Mail Stop 262-4 Moffett Field, CA 94035-1000, USA

Abstract

This paper describes how intelligent agents can simulate human operators to aid in the analysis and design of complex systems. The paper presents two examples of adapting the Crew Activity Tracking System (CATS) to function as an intelligent agent. The first is a model-based design application, in which CATS agents perform the task of air traffic controllers in order to test a new operational concept. The second concerns human error analysis, in which a coordinated team of CATS agents represents a flight crew and aircraft destroyed due to controlled flight into terrain. The paper also discusses issues regarding the development of task model-based agents that make plausible, human-like errors and the use of such agents.

Keywords
Agents, analysis and design, air traffic control, aviation, human performance models

1 Introduction

Applications of software agents—intelligent, autonomous computer code—are proliferating as agent technologies advance. One application of particular interest for the analysis and design of complex systems is to use agents to simulate human operators. Agents can 'close the loop' between other human operators and/or automated systems to provide a holistic view of system operations. The more complex and team-oriented a system is, the more expensive and time consuming it is to conduct human-in-the-loop simulations of the entire system. Furthermore, if the operational concepts to be tested are especially novel, it may be difficult to recruit and train humans that can skillfully manage the task and remain free from biases that stem from experience with present day operations. Agents, on the other hand, can perform the task in specific, controllable ways to test whether a particular technique offers specific advantages, or whether it creates problems for other subsystems. Such agent-based simulation studies can determine if the concept works well enough to warrant full-mission human-in-the-loop simulation. If it does, these agent studies can determine how the human subjects should be trained.

This perspective on simulating human operators differs from that proffered by cognitive modeling researchers (e.g., [1]). The emphasis, at least initially, was on testing and enhancing the fidelity of models of human perception and cognition by implementing the models in computational agents. More recently, researchers sought to incorporate simulated humans into the design process to assess the design with respect to not only the physical attributes of human operators, but also the cognitive attributes (e.g., [11]).

Another important line of research emphasizes the importance of operator task models to support the design process, and the potential for using these models to develop operator training and aiding systems (e.g., [13]. Yet another uses task analysis as a tool to help ensure that designs enable operators to directly perceive functional constraints in the task environment to support task performance (e.g., [12, 17]). Increasing use of distributed simulation techniques for design has made the benefits of using simulated human agents all the more attractive, and so-called 'human performance models' have broadened in scope to include agents based on both task models (or 'activity models'), and cognitive models (e.g., [9]).

This paper describes the Crew Activity Tracking System (CATS) as a framework for developing intelligent agents. CATS originated, however, as a task model–based framework to support intelligent training and aiding applications in complex systems [6, 7]. Indeed, proof-of-concept training and aiding systems show promise (e.g., [3]), but to date CATS has yielded more tangible benefits as a tool to support the design and analysis of operator procedures for new operational concepts [2, 4, 8]. Visualization capabilities have been added to CATS to bolster its effectiveness in this capacity [2, 4, 5].

Simulating human operators is not always addressed as an integral part of a model-based design process (cf. [13]). But while CATS fulfills requirements for supporting a model-based design process that culminates with operator assistance, CATS can also be adapted into a human performance model. To help design new systems, 'CATS agents' can simulate proposed operator roles to test the behavior of other system components and investigate the error tolerance of the overall system. For existing systems, CATS agents can simulate operator behavior using knowledge of current operating procedures to shed light on 'error chains' that contribute to unsafe situations. This paper briefly describes CATS, then presents two examples of using CATS agents for analysis and design. Finally, it discusses the development of task model-based agents that make plausible, human-like errors, and the potential uses of such agents.

2 Crew Activity Tracking System

The original CATS implementation uses a computational task model to 'understand' the activities the crew performs when navigating using the automatic flight modes found on the Boeing 757 aircraft, in order to support training and aiding for mode awareness [6]. CATS takes as input air traffic control clearances received by the aircraft via data link, and data about the aircraft and onboard system states, and uses its model to predict how the crew should preferably configure the autopilot in order to comply with a clearance. As pilots perform actions, CATS uses its model to check that the operations are performed correctly. In this sense, CATS 'tracks' flight crew activities to 'understand' that they are error-free.

The activity tracking process has two threads: one predicts preferred activities, and one interprets actual operator actions. Fig. 1 depicts the CATS architecture that implements the activity tracking methodology. As the state and constraint representations that comprise the data model are updated, CATS correspondingly updates 'context specifiers' that comprise the current context model. CATS then searches its operator activity model to find the activities that have conditions for prediction that the current context satisfies. When an activity is predicted, CATS starts a timer and waits for the operator to execute the activity. The second thread of the process, the action interpretation thread, attempts to interpret operator actions by linking them to the predicted activities and, failing that, to acceptable alternatives. Actions that CATS cannot interpret ('uninterpretable' actions) may be errors. Possible errors of omission are signaled when a timer expires before an operator performs a predicted or alternative valid action. Details about the activity tracking process, an example CATS model, and a validation study are provided in [7]; enhancements and further applications are discussed in [2, 3, 4, 5, 6, 8].

3 CATS Agent Development

Two approaches have been pursued for using CATS as a framework for developing human performance models. In the first, a CATS agent simply 'executes' the actions identified by the prediction thread of the activity tracking process. The second approach uses a more elaborate scheme for coordinating the activities of multiple CATS agents—including cognitive and

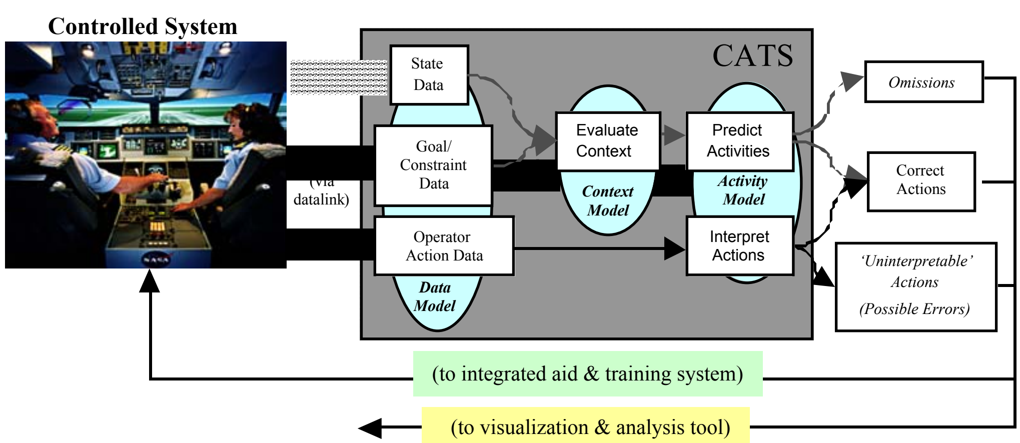

Fig. 1: CATS knowledge representation and processing scheme, showing input from a controlled system, and activity tracking output and applications.

perceptual activities. The next two subsections describe these approaches; examples of their use follow.

Nominal Agent

In the 'nominal agent' scheme, a nominal CATS model of proposed operator tasks required for a new operational concept is developed as prescribed for model-based design. CATS links to a distributed simulation that includes proposed automation, other operators, etc. It requests and receives data, identifies the current context, and uses the model to predict activities (Fig. 2). Instead of waiting for the operator to perform actions, so it can track them, the agent executes them in the simulation loop. To simulate activities that are not 'instantaneous,' the CATS model can incorporate terminating conditions to reflect the duration of activities. Under this scheme, a CATS agent may use multiple models of individual operators, or a single model that represents the tasks of the entire team, annotated to capture individual responsibilities. Also, a single CATS agent or multiple agents may be in the simulation loop.

Fig. 2: CATS simulation of nominally predicted activities.

Coordinated Teams of Agents

A second approach produces agents that can coordinate their activities with other agents. This more radical adaptation uses activities to represent aspects of cognition (cf. [18]). Instead of simply operating according to the true state of the world, as in the 'nominal agent' scheme, the agent must actively 'perceive' aspects of the environment by performing activities that represent perceptual processing and, similarly, actively 'recall' this information for use. Instead of formulating the current context based on the true state of the world, each agent maintains 'beliefs' about the world that may or may not be true (cf. [10]). The CATS activity representation of 'cognitive,' manual, verbal, and 'perceptual' tasks related to monitoring, assessment, communication, and control guides the way in which an agent acquires, transforms, and transfers beliefs to other agents. The transferal of beliefs provides the mechanism through which multiple agents of this form can coordinate.

This scheme is depicted in Fig. 3. The agent performs monitoring activities to 'perceive' beliefs from other agents (e.g., by hearing what they verbalize), or states and constraints from the environment (not shown). The agent converts perceived information to beliefs via assessment activities; assessment activities also transform existing beliefs by comparing them, aggregating them, etc. The set of beliefs that results from this process represents the 'perceived context' (as opposed to the true context). The agent then enables activities based on the perceived context. However, if the activity involves coordinating with another agent, the agent does not actually execute the activity until the other agent also has beliefs that allow the 'other side' of the coordination to occur. The CATS activity model represents all of the information required to specify the conditions (in terms of rules made up of beliefs) under which the agent should enable an activity. The model also represents what beliefs the agent acquires, transforms, or transfers to which other agent(s) when the agent performs the activity (again, if the other agent(s) are prepared to coordinate).

A simple processing scheme is implemented in a 'CATS Agent Simulation Executive.' The executive uses a 'pool' of beliefs that agents are attempting to transfer during that current processing cycle as the medium for coordination between agents (multiple distributed executive processes for agents of this form are beyond the scope of this paper). The executive first enables activities from each agent model and places them in queues according to class (i.e., cognitive, manual, perceptual, verbal). It then attempts to execute each activity in each queue of activities. Contrary to the 'flow' implied in Fig. 3, the executive executes manual and verbal activities first during each cycle, so that beliefs get added to the 'pool' and made available to other agents. Next, agents perform cognitive activities to transform beliefs, if necessary. Finally, agents

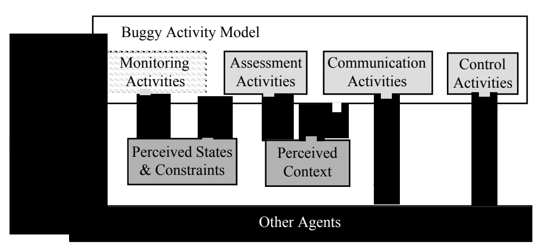

Fig. 3: CATS agent acquires and transforms beliefs via monitoring and assessment activities, and coordinates with other agents by transferring beliefs via communication and control activities.

perform perceptual activities to acquire beliefs from the 'pool' that are there as a result of activities performed by other agents. Effective task coordination requires that some beliefs pertain to what other agents are doing, or to what the agent needs to remember to do itself (i.e., prospective memory). This scheme allows agents to coordinate using models of arbitrary complexity.

Fortuitously, the Java™ implementation of the CATS model enables these 'cognitive' agents to be developed with relatively little additional code. The executive process requires usual CATS queries to the model, and the beliefs processed when an activity is executed are represented through reuse of the CATS model specification syntax. A symbolic notation allows agents to locate relevant beliefs in the 'pool' and keeps additional processing code to a minimum.

4 Example I: Air Traffic Controller Agents for Design

The first example concerns research in which CATS agents simulated the performance of air traffic controllers (henceforth, 'controllers') to support model-based design of a new air traffic management concept. One of the initial motivations developing controller agents was to supplant scarce and expensive human controller subjects in the early phases of developing a large-scale distributed simulation of the new operational concept. Also, the research could better explore issues surrounding changes to information displays and interaction methods, by capturing key features of the interactions between controllers and controlled aircraft. Such features include clearance types and the pace at which controllers issue them given various attributes of the traffic flow [15]. For example, the ebb and flow of workload imposed on controllers could be examined via a timeline showing clusters of different classes of activities (Fig. 4).

These agent simulations were developed using the 'nominal agent' approach. Agents represented two interacting controllers, each with a separate model reflecting their distinct roles. CATS performed the basic process of assessing current context, then predicted (i.e., executed) controller activities according to both models. Some contexts could result in both agents performing actions. Using these agents, a new interface prototype was developed to provide the information needed for the new operational concept.

A second phase of the research [16] sought to estimate benefits that could be realized under the proposed concept. One potential benefit is a reduction in the number of clearances that controllers must issue. Radio frequency congestion and delayed radio transmissions (i.e., transmissions that cannot be executed immediately because there is already another one in progress) are areas of concern. Actual controller data for the controller positions and airspace in question were used to enhance the

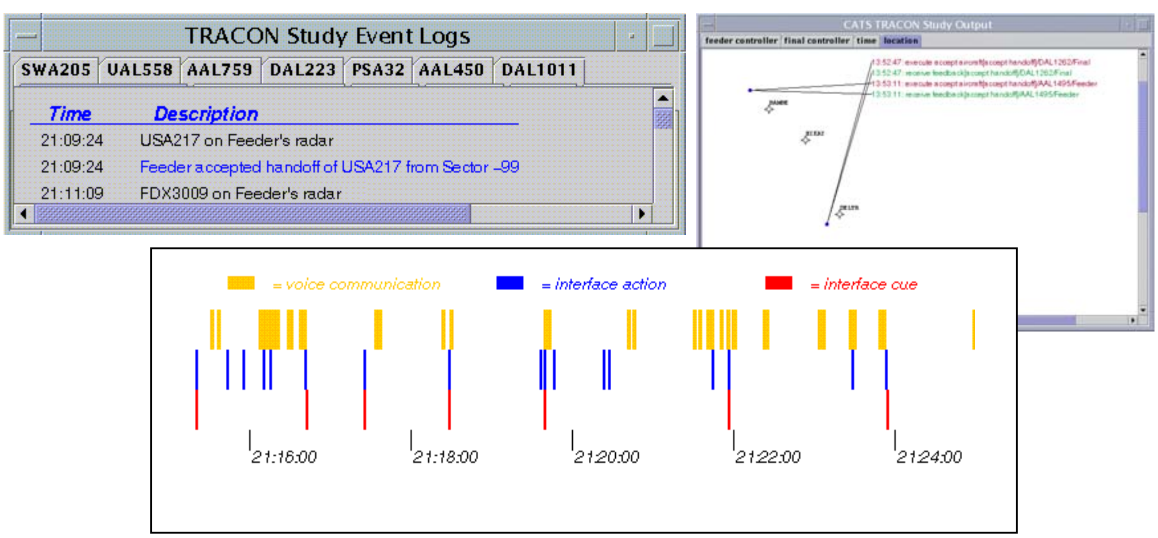

Fig. 4: Sample output from a CATS agent simulation of an air traffic controller, showing salient events, locations of aircraft when the agent performed actions on them, and a timeline of activities.

CATS agents with terminating conditions based on the average duration of a particular task (e.g., issuing a particular type of clearance). This set of models was then further modified to exclude all the activities that, in the ideal case, would be unnecessary under the new operational concept. Attributes of the associated traffic data were also changed so that plausible numbers of the aircraft represented in the traffic data would be appropriately equipped and operating under the new concept.

The results of the agent simulations provided estimates of the maximum possible benefits that could be realized under the proposed concept: a forty per cent reduction in the number of transmissions required per aircraft is possible. The simulations also afforded the opportunity to examine the phenomenon of delayed radio transmissions. Currently, when workload is especially high during arrival rushes, a transmission has to be delayed approximately every thirty seconds. The agent simulations indicated possible improvements of ten to twenty per cent (depending on specific circumstances). Thus, in addition to aiding in the identification of information requirements and new interaction methods, the CATS agent simulations helped quantify potential benefits of the air traffic management concept. Incorporating agents into this particular model-based design effort provided insights that would have been much more costly to obtain through human-in-the-loop simulations.

5 Example II: Pilot Agents for Accident Analysis

Another research effort used the ‘coordinated teams of agents’ approach described above to recreate a controlled-flight-into-terrain (CFIT) aircraft accident. Organizers of a FAA/NASA Aviation Safety Program workshop on human error solicited agent-based models of the accident for presentation. The accident involved several factors known to increase the likelihood of crew errors, including a ‘non-precision’ approach, bad weather, and a fatigued crew; non-precision approaches in particular are often associated with CFIT accidents.

A simulation with coordinating CATS agents was developed with the architecture shown in Fig 5. Two agents represent a human crew (Pilot Flying, and Pilot Not Flying), and a third represents the aircraft’s cockpit interface (‘Aircraft’ in Fig. 5). Treating the aircraft

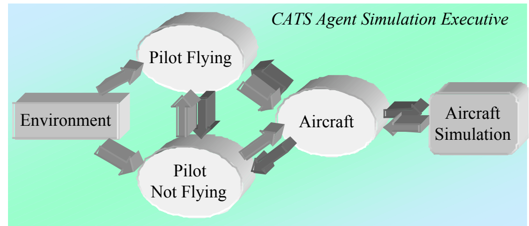

Fig. 5: Coordinated CATS agent simulation; ovals represent CATS agents.

displays and controls as an agent allows the pilot agents to ‘tell the aircraft’ what current target values are, and the aircraft can ‘tell the crew’ about its state. The aircraft agent is linked to a simulation that updates the aircraft’s state on each processing cycle.

Models of the tasks each pilot nominally performs when flying a non-precision approach procedure were adapted from previously validated CATS models [7] and training manuals, and used as the starting point for developing the CATS agents. For the pilot agents, tasks such as "maintain situation awareness," "maneuver aircraft," "configure aircraft for landing," and "perform approach communications" were decomposed into cognitive, verbal, manual, and perceptual actions. The specific beliefs that are acquired, transformed, or transferred to another agent when the agent performs a given action were also added to each model (e.g., "current_alt _is_ 1980 feet," "target_alt _is_ 1400 feet"). The aircraft agent was modeled to ‘perform tasks’ such as "update target altitude," to represent receiving a new target altitude when the pilot-not-flying set one, and "display altitude" to represent supplying the current altitude to the pilots when they perform "monitor altitude" actions. The aircraft ‘tasks’ were modeled as always enabled, so that the other agents could interact with the aircraft whenever they needed to.

The agents were implemented to create the simulation shown in Fig. 6. According to the processing described above, each cycle began by checking whether an environmental event, such as an air traffic control clearance, had occurred. Then enabled activities were added to each agent’s activity queues, and the process of interacting with the ‘pool’ of beliefs was performed. A series of iterative model enhancements resulted in a simulation that could execute the approach procedure, including relevant "call-outs," and other coordination activities, and successfully land the aircraft.

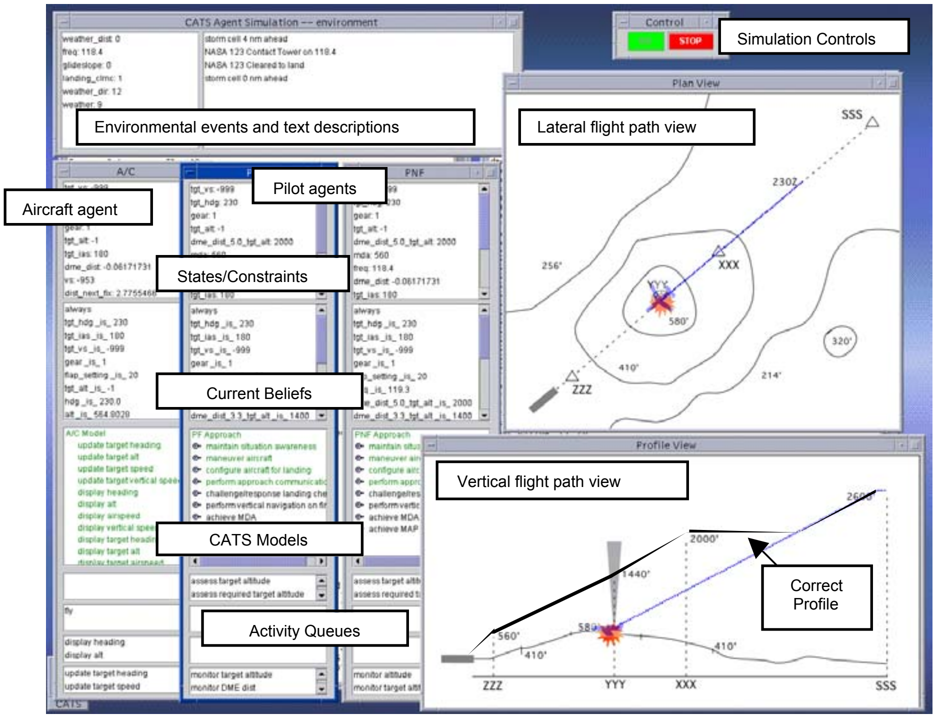

Fig. 6: Screen snapshot of the CATS agent CFIT accident simulation, with key components identified.

An analysis of the required activities, and the beliefs each agent had to have when in order to execute them, indicated that the accident trajectory could be explained if the pilots had misread their approach chart in a specific way. For the non-precision approach in question, the Distance Measuring Equipment (DME) distance used to specify when the aircraft should ‘step down’ to the next lower altitude was measured not from the runway threshold, but from a point several miles before it. By initializing the set of beliefs of the pilot agents to reflect the incorrect DME distances, the simulation culminates with a CFIT accident at almost the precise location relative to the terrain as the real accident upon which the scenario was based (Fig 6).

These results indicate that this non-precision approach chart may have contributed to the crew error. However, a number of unanswered questions remain regarding how the crew failed to catch and correct their error, given the number of activities they performed that are designed to check and double-check for such dangers. In particular, the warning system that indicates proximity to terrain was working properly, and modeled in the agent simulation. However, including the nominally correct response to the alert ("Pull up! Pull up!") in the pilot models averts the CFIT accident.

6 Discussion

The examples above demonstrate that CATS-based agents can be useful for analysis and design of complex systems. There are also numerous intriguing future lines of investigation, including one hinted at in Fig. 3: ‘bugs’ in the model of activities, mistaken beliefs, or slips introduced via mechanisms thought to contribute to such errors in humans [14] could be added to create agents that err in realistic ways. Such agents could be used to investigate the error tolerance of a system in a manner similar to that in which the ‘nominal agent’ approach is used to test how the system operates properly. Additional research is needed, however, to adequately represent error mechanisms and, equally important, error recovery mechanisms, as they relate to system operations.

A parallel problem is that errors lead to disasters only infrequently. How can simulations produce error chains for which a system is most at risk in a short time? CATS agents may help, because they can be implemented reasonably quickly. Such agents, in conjunction with formal system models, may help focus the effort on aspects of operations that are particularly susceptible to errors. Thus, instead of relying on

Monte Carlo simulations, and hoping to stumble across an error situation worth investigating further, agents could err and fail to recover in systematic ways, and any negative effects on overall system operation could be flagged. By further restricting the analysis to portions of the operating regime identified as unsafe via formal methods, analyses could perhaps address risks to safe overall operations before the design is fielded.

7 Acknowledgements

This research was supported by the Human Automation Integration Research element of NASA's Aerospace Operations Systems R&T Base Program, and by the Human Error Modeling element of the FAA/NASA Aviation Safety Program.

References

[1] J. Anderson, The Adaptive Character of Thought, Hillsdale, NJ: Lawrence Erlbaum, 1990.
[2] T. Callantine, A Glass Cockpit Crew Activity Analysis Tool, SAE Technical Paper 200-01-5522, Warrendale, PA: SAE International, 2000.
[3] T. Callantine, An Intelligent Aid for Flight Deck Procedure Performance: The Crew Activity Tracking System "Task Caddy," Proc. of IEEE Conference on Systems, Man, and Cybernetics, Tokyo, pp. 1095-1100, 1999.
[4] T. Callantine, Online Analysis Of High-Fidelity Simulation Data, Proc. of 2000 Conference on Cognitive Systems Engineering in Process Control, Taejon, Korea, pp. 175-181, 2000.
[5] T. Callantine and B. Crane, Visualizing Pilot-Automation Interaction, In K. Abbot, J. Speyer, and G. Boy (Eds.), HCI-Aero 2000 International Conference on Human-Computer Interaction in Aeronautics, Toulouse: EURISCO, pp. 87-92, 2000.
[6] T. Callantine and C. Mitchell, A Methodology and Architecture for Understanding How Operators Select and Use Modes of Automation in Complex Systems, Proc. of IEEE Conference on Systems, Man, and Cybernetics, San Antonio, TX, pp. 1751-1756, 1994.
[7] T. Callantine, C. Mitchell, and E. Palmer, GT-CATS: Tracking Operator Activities in Complex Systems, NASA Technical Memorandum 208788, Moffett Field, CA: NASA Ames Research Center, 1999.
[8] T. Callantine, E. Palmer, and N. Smith, Model-Based Crew Activity Tracking for Precision Descent Procedure Refinement, Proc. of IEEE Conference on Systems, Man, and Cybernetics, Orlando, FL, pp. 3126-3131, 1997.
[9] G. Campbell and J. Cannon-Bowers, Human Performance Model Support for a Human-Centric Design Process, SPIE Aerosense 2000, San Diego, CA, August, 2000.
[10] W. Clancey, P. Sachs, M. Sierhuis, and R. van Hoof, Brahms: Simulating Practice for Work Systems Design, International Journal of Human-Computer Studies, Vol. 49, pp. 831-865, 1998.
[11] K. Corker, and G. Pisanich, Analysis and Modeling of Flight Crew Performance in Automated Air Traffic Management Systems, Proc. of 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, Cambridge, MA, 1995.
[12] A. Kirlik, A., Requirements for Psychological Models to Support Design: Toward Ecological Task Analysis, In J. M. Flach, P. A. Hancock, J. K. Caird, and K. J. Vicente (Eds.), An Ecological Approach to Human Machine Systems I: A Global Perspective, Hillsdale, NJ: Lawrence Erlbaum, 1995.
[13] C. Mitchell, Model-Based Design of Human Interaction With Complex Systems, In W. B. Rouse & A. P. Sage (Eds.), Handbook of Systems Engineering & Management. New York, NY: John Wiley & Sons, 1999.
[14] D. Norman, The Psychology of Everyday Things, New York: Basic Books, 1988.
[15] S. Romahn, T. Callantine, and E. Palmer, Model-Based Design of Air Traffic Controller-Automation Interaction, Proc. of IEEE Conference on Systems, Man, and Cybernetics, La Jolla, CA, 1998.
[16] S. Romahn, T. Callantine, and E. Palmer, Model-Based Design and Analysis of ATC-Automation Interaction, Proc. of 10th International Symposium on Aviation Psychology, Columbus, OH, pp. 640-645, 1999.
[17] K. Vicente, Cognitive work analysis: Toward Safe, Productive, and Healthy Computer-Based Work, Mahwah, NJ: Erlbaum, 1999.
[18] W. Zachary, J. Ryder, and J. Hicinbothom, Cognitive Task Analysis and Modeling of Decision Making in Complex Environments. In J. Cannon-Bowers & E. Salas (Eds.), Making Decisions Under Stress: Implications for Training and Simulation, Washington, DC: American Psychological Association, 1998.