Schoelles, M. J., & Gray, W. D. (2011). Cognitive modeling as a tool for improving runway safety.
Dayton, OH: 16th International Symposium on Aviation Psychology (ISAP).

COGNITIVE MODELING AS A TOOL FOR IMPROVING RUNWAY SAFETY

Michael J. Schoelles
Wayne D. Gray
Rensselaer Polytechnic Institute
Troy, New York

Runway incursions are low probability events resulting from complex combinations of cognitive and environmental factors, which can have deadly consequences. However, the development and evaluation of tools to reduce runway incursions are, ironically, hampered by the low incidence of such events. A possible path forward is the use of high-fidelity cognitive models to predict pilot performance under a wide variety of airport conditions and runway circumstances. We describe a fully embodied ACT-R 6.0 cognitive model, named SimPilot, of a pilot taxiing a simulated Boeing 737-800 aircraft. The goals of SimPilot are twofold. The first is automated testing of a new safety devices. The second goal is to show that modeling the multitasking inherent to taxiing in a cognitive plausible manner is an important step in predicting and preventing runway incursions.

New tools are continually being developed to increase aviation safety by reducing human error. Since it has been shown that small changes in the design of a system can lead to large changes in human performance, imagine the potential changes that could occur as the result of introducing a new tool. To ensure that the changes introduced are beneficial, extensive testing with subject matter experts (SMEs) is required, but in the aviation domain this is expensive. A potential solution to this problem is simulating SMEs with cognitive models. The type of models that can provide useful simulations are process models that can actually do the task and are based on a cognitive architecture.

In this paper we describe a cognitive model to test the Electronic Movable Map (EMM) tool that is intended to improve runway safety. Note the EMM used in this effort is not the real EMM developed for NASA but a functionally equivalent software version developed for testing this methodology. The cognitive model works in conjunction with the Aptima developed Performance Engine (PE), which collects and stores data from a data source and computes relevant performance data. To test the EMM, scenarios were developed to create runway incursions. The scenarios are to be followed by human pilots or model simulated pilots taxiing a Boeing 737 with and without the aid of an EMM.

Taxiing requires many cognitive and perceptual/motor interactions. For example, the pilot must steer the plane, monitor the taxiing speed, listen to directions from Air Traffic Control, watch for other aircraft, etc. Therefore a model of the pilot must be able to multitask. Recently, multitasking has received a lot of attention in the press and has become a research area for Cognitive Science and in particular, cognitive modeling. We choose ACT-R 6.0 as the cognitive architecture for this project in part because it supports multitasking. The version of ACT-R that we use incorporates an add-on that implements the Threaded Cognition theory (Salvucci & Taatgen, 2008, 2011) of multitasking. An additional reason for using ACT-R in this effort is the success by Byrne (Byrne & Kirlik, 2005) in modeling the taxiing task using ACT-R 5.0.

We use the X-Plane 9 Desktop Simulator™ that is available for MAC, Windows, and Linux operating systems. It supports an easy to use plug-in capability to connect the simulator to data collection systems and aircraft control systems. In the next section we describe the parts in more detail and show how all the parts fit together.

System Configuration and Operation

Figure 1 shows the configuration of the varied components in this test bed. There are five computers connected though a Network Switch.

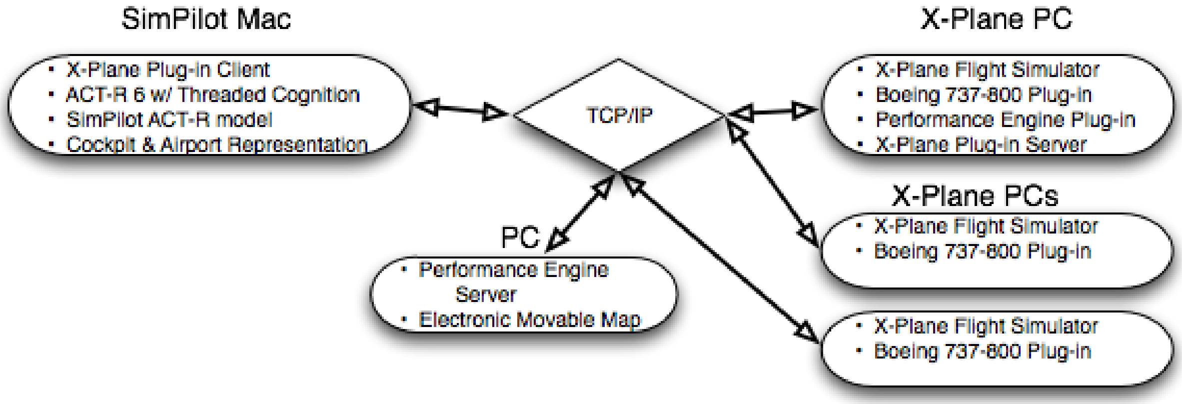


***Figure* 1. The SimPilot Mac, X-Plane PC, and Electronic Movable Map PC and two additional aircraft connected through a TCP/IP Switch.**

For the model to be able to simulate a pilot, it must be able to *see* what the pilot sees, *hear* what the pilot hears, and *manipulate* the aircraft controls. The Cockpit and Airport Representation modules enable these functionalities by providing a representation of the cockpit instruments and airport layout.

The airport representation module is constructed from data sent by the EMM. The model manipulates the aircraft controls by sending joystick commands, mouse movements and clicks to the Model Server running on the X-Plane PC. It receives data about the cockpit instruments from the Model Server. Figure 2 shows the cockpit.

***Figure* 2. The Boeing 737-800 cockpit.**

The SimPilot Mac also contains the scenario script, which it is to follow. The EMM and Performance Engine receive data about the Boeing 737 plane and other planes at the airport through the Performance Engine Server. The EMM sends aircraft positions and airport layout to the Airport Representation on the MAC. In this way the SimPilot achieves the functionality to see other airplanes, taxiways, and runways.

Model Description

The heart of our effort, SimPilot, is a work in progress. Most of the effort to date has gone into the development of the vitally necessary and excruciatingly detailed Cockpit and Airport Representations. However, our initial integration testing with the other components has been successful. In this section we will first describe the cognitive architecture on which the model is built. We will next describe the major change to the architecture required to perform the level of multitasking necessary for taxiing. The details of the model structure are then presented and the section ends with a discussion of the limitations and problems of our current approach.

ACT-R 6.0

ACT-R 6.0 (Anderson et al., 2004) is an embodied cognitive architecture that has perceptual and motor components along with cognitive processing, memory, and control components. The perceptual and motor components enable SimPilot to operate user interfaces by passing the interface software the same commands passed by the input devices used by humans. As the SimPilot is a cognitive, not an artificial intelligence model, its input commands mimic the speed and accuracy of human users.

In common with all ACT-R models, SimPilot consists of pattern matching and action rules. When a match is found, the action associated with that pattern is executed. ACT-R executes one rule at a time. ACT-R maintains simulated human time in that time for ACT-R processes and actions are set to the theoretical times for the corresponding human events. For example, to shift visual attention from one object to another takes 85 ms. Time to retrieve an item from memory varies as a function of the recency and frequency of that item's occurrence (Schooler & Anderson, 1997; Sims & Gray, 2004). When the model does a task ACT-R produces a trace that includes the action taken and a time stamp. The trace allows model performance to be compared with human performance.

ACT-R checks every 50ms (human time) all of its rules and executes one of the rules whose pattern is matched. If more than one rule can execute then ACT-R chooses the one it calculates would be the most useful at this time. As ACT-R makes this decision every 50 ms, this serial execution is not as constraining as it might seem and has been shown to be as accurate at simulating fine-grained human behavior as architectures that allow parallel firing of rules (Byrne & Anderson, 2001). If a rule could fire but did not because another one had a higher utility, chances are that in 50 ms it will be able to fire. The rules are intended to represent the fine-grained procedural steps that are executed to perform some task. As biological processes are inherently noisy in the signal processing sense of noise (Faisal, Selen, & Wolpert, 2008; Neri, 2010), ACT-R adds noise to the utility calculation to simulate the variability in time and performance that humans make. Besides a procedural memory ACT-R has a declarative memory that holds units or *chunks* of factual information. These chunks represent the portion of the simulated human's background knowledge necessary to perform the task that the model is attempting to perform. Like humans, errors can occur in the memory retrieval process due to random fluctuations (noise) in memory strength or activation (Sims & Gray, 2004). Either the wrong chunk is retrieved or the intended chunk is not "strong" enough to be remembered.

The perceptual components of ACT-R allow the model to see and hear. In common with the human brain, the visual component has *where* and *what* paths (Findlay & Gilchrist, 2003). The *where* path allows the model to detect features of an object such as color, size, and shape at a 2-D location in space. The *what* path moves visual attention to that location to encode the object with those features. ACT-R *hears* in much that same way that it sees in that sound events are detected and auditory attention is invoked to encode those sounds. By encoding objects and sounds in the environment the visual and auditory components add new declarative knowledge to the model. The motor component is the model's *hands* and *voice*. The manual component is capable of moving and clicking the mouse. Movement times are a based on Fitts' Law (Fitts, 1954). The vocal module is capable of speaking text and subvocalization (see, e.g., Huss & Byrne, 2003).

The *imaginal* component of ACT-R is intended to hold intermediate representations required in solving a problem or performing some task. New declarative chunks can be added by this component. The *temporal* component maintains an internal clock. The *goal* component in hold chunks that guide task execution. For the model presented in this paper the default goal component is replaced with a module that implements a form of Threaded Cognition (Salvucci & Taatgen, 2008) that implements the multitasking required for the taxiing task.

Threaded Cognition

Salvucci and Taatgen (Salvucci & Taatgen, 2008, 2011) propose Threaded Cognition as an integrated theory of concurrent multitasking. Multitasking is defined as doing 2 or more tasks at once. A thread is sequence of processing steps coordinated by a serial procedural resource and executed across perceptual and motor resources. The key claims of Threaded Cognition are that multiple active goals can exist. Associated with each goal is block of procedural processing. Processing conflicts can exist for procedural, declarative, perceptual, and motor resources. A thread will grab a resource if it needs the resource and the resource is available. It will release the resource when no longer needed. According to Salvucci and Taatgen, cognition favors the least recently processed thread. Declarative retrievals can be converted to hard coded rules over time thus reducing both declarative and procedural resource conflicts. The cognition requires no central and supervisory executive.

Most of the points in the paragraph above have been part of ACT-R since version ACT-R 5.0, thus the implementation of Threaded Cognition into ACT-R 6.0 requires only allowing two or more active goals and giving priority to the least recently processed goal. These changes are implemented in the version of ACT-R 6.0 used by our model by simply using a different goal component. Threaded Cognition is a relatively new theory and has not been tested in complex, dynamic environments. Also it has rarely been tried on more than two tasks.

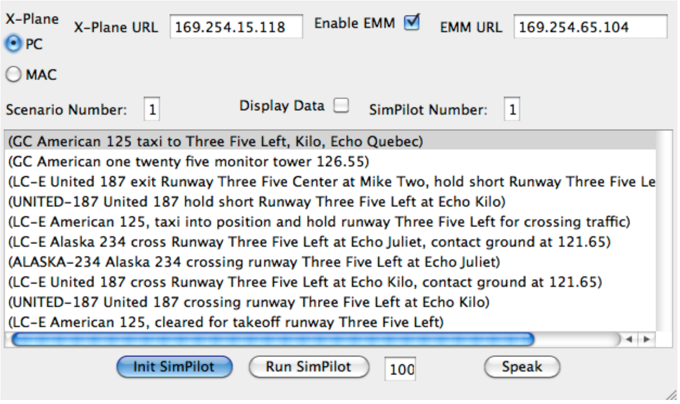
Model Processing

In this section we try to give a flavor of how the model operates and some of the other problems involved. The taxiing task is divided into subtasks. Each subtask is represented as a chunk, which has (a) an initial state, (b) the set of treads that can run simultaneously in the subtask, and (c) a final state. In general the final state of a subtask is the initial state of another subtask. The subtask chunks are put in the goal buffer and their associated productions initiate the threads specified in the chunks. In general the state of the other resources control the execution flow in the spirit of Taatgen’s (2007) Minimal Control Principle.

The basic actions that the model must perform include setting switches, tuning radios and monitor indicators, working the throttle, monitoring aircraft speed, steering the aircraft, turning the aircraft, stopping the aircraft, listening to air traffic control, and watching for other aircraft.

All scenarios begin with preflight checks completed and the aircraft sitting on the tarmac waiting for instructions from Air Traffic Control (ATC). The execution of a scenario that provides ATC instructions requires experimenter action. The scenario text is shown in a window on the MAC screen, see Figure 3, and when a scenario event is to be executed it is clicked on. For example, to start the first scenario, the experiment would highlight the text "American 125 taxi to Three Five Left, Kilo, Echo Quebec" and the click the execute button. This causes a sound event within ACT-R. The model contains a rule that whose pattern simply matches any and all sound events and ignores any other environment or internal event. So when the event does occur a processing thread is initiated to attend and encode the sound. The sound is interpreted as the command to begin taxiing which sets a goal to execute the begin-taxiing-procedure that increases engine thrust, until the engines are at the proper N1 level and then releasing the parking brake. Increasing thrust and monitoring the N1 level is an example of the coordination required between manual actions and visual monitoring that is difficult to model. One reason is that it is difficult is the display of the N1 levels lags behind the manual action. In general, the model tends to overshoot and must make a series of corrective actions. Real pilots have a “feel” for doing this that which is beyond the current state-of-the-art in cognitive architectures.

To perform an action on an instrument control, such as releasing the parking brake and checking its state, the model must be able to see the control. Seeing is done through the cockpit representation, called the *visicon*. The visicon contains 77 entries, one for every light, switch, dial, lever, and display in Figure 2. For each entry, the visicon contains its location on the display, its size, its color, its value current value, and its type. These are the details that allow the vision module to function as if it sees these items. The visicon must be updated whenever cockpit display changes otherwise the model will not see the results of its action. Achieving the integration of these two software systems was a significant software engineering challenge as the X-Plane software only provides a third of the 77 item locations needed by the model. For the rest they must be manually calculated offline, which is not desirable since this is time-consuming and they may change with new versions of flight simulator. In addition, if the entire cockpit display can't fit on the screen then the locations in the visicon must be adjusted when the screen scrolls. This is problematic because the scrolling action is purely an interface issue and not relevant to the taxiing task.



X-Plane
● PC
○ MAC
X-Plane URL 169.254.15.118
Enable EMM ☑
EMM URL 169.254.65.104
Scenario Number: 1
Display Data ☐
SimPilot Number: 1

(GC American 125 taxi to Three Five Left, Kilo, Echo Quebec)
(GC American one twenty five monitor tower 126.55)
(LC-E United 187 exit Runway Three Five Center at Mike Two, hold short Runway Three Five Le
(UNITED-187 United 187 hold short Runway Three Five Left at Echo Kilo)
(LC-E American 125, taxi into position and hold runway Three Five Left for crossing traffic)
(LC-E Alaska 234 cross Runway Three Five Left at Echo Juliet, contact ground at 121.65)
(ALASKA-234 Alaska 234 crossing runway Three Five Left at Echo Juliet)
(LC-E United 187 cross Runway Three Five Left at Echo Kilo, contact ground at 121.65)
(UNITED-187 United 187 crossing runway Three Five Left at Echo Kilo)
(LC-E American 125, cleared for takeoff runway Three Five Left)

100

***Figure* 3. SimPilot and Scenario Control Screen**

The procedure that SimPilot uses to turns the aircraft is based on a task analysis generated a pilot. This analysis specifies speed changes and thrust changes for turning. An important data point in turning is when to start to turn. A pilot can see the intersection and through experience knows when to start turning. The model does have a representation of the airport that it acquires from the EMM, and it does have knowledge about where it is on the ground so the distance to the turn can be calculated, at this time it is not known whether this will be of sufficient fidelity to simulate real turns.

Human Performance Modeling in Aviation

In 2005, NASA chose five teams to develop human performance models (HPM) of pilots performing taxi operations with and without advanced displays (Foyle et al., 2005). Each team used a different modeling architecture. The Attention-Situation Awareness approach by Wickens and McCarley (Wickens et al., 2005) looked at attention allocation and situational awareness. Simulation data drove a model that predicted errors and the benefits of the T-NASA display. The ACT-R 5.0 model by (Byrne, Kirlik, & Fleetwood, 2008) is the closest to the model presented here. They were connected to an X-Plane Simulator but concentrated on decision-making strategies rather than multitasking because of the limitations of ACT-R at the time. Air-MIDAS by Corker (Foyle, et al., 2005) used working memory limits, interference processes and heuristics to predict errors. D-OMAR (Foyle, et al., 2005) by Deutsch and Pew is an event-based simulator with three different languages to develop perceptual, cognitive and motor processes, which they considered to be the building blocks of pilot expertise. They found that because errors are so infrequent, habit might intrude and lead to certain types of errors. IMPRINT by Lebiere and Archer (Foyle, et al., 2005) combined IMPRINT, which is a performance tool with ACT-R. IMPRINT provided the simulation environment and ACT-R acted as cognitive agent.

Conclusions

We have described a multitasking cognitive model based on the ACT-R architecture. The goals of the model are both applied and theoretical. On the applied side, we hope to advance the methodology of automated testing using simulated human experts rather than using actual people. This form of testing can be beneficial in testing design changes, particularly major changes that involve adding new tools and technology to an already proven system. On the theoretical side we are pushing the current theory in multitasking in order to indentify its weaknesses and what further changes to the architecture are required to make this a more robust theory.

Acknowledgements

This work was supported by NASA Ames Research Center through a Small Business Innovative Research Contract (Contract No. NNX09CB45C) awarded to Aptima, Inc. All opinions expressed in this article are those of the authors and do not necessarily reflect the official opinion or position of Rensselaer Polytechnic Institute, Aptima, Inc., or NASA. We would like to thank Dr. James Johnston and Dr. Michael Feary at NASA Ames Research Center for funding this research. We also would like to thank Gabe Ganberg, Mike Therrien, Joe Thoreson and Jamie Estock of Aptima, Inc.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. L. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060.

Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychological Review, 108*(4), 847–869.

Byrne, M. D., & Kirlik, A. (2005). Using computational cognitive modeling to diagnose possible sources of aviation error. *International Journal of Aviation Psychology, 15,* 135-155.

Byrne, M. D., Kirlik, A., & Fleetwood, M. D. (2008). An ACT-R approach to closing the loop on computational cognitive modeling: Describing the dynamics of interactive decision making and attention allocation. In D. C. Foyle & B. L. Hooey (Eds.), *Human performance modeling in aviation* (pp. 77-104). Boca Raton, FL: CRC Press.

Faisal, A. A., Selen, L. P. J., & Wolpert, D. M. (2008). Noise in the nervous system. *Nat Rev Neurosci, 9*(4), 292-303.

Findlay, J. M., & Gilchrist, I. D. (2003). *Active vision: The psychology of looking and series*. New York: Oxford University Press.

Fitts, P. M. (1954). The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology, 47*(6), 381–391.

Foyle, D. C., Hooey, B. L., Byrne, M. D., Corker, K. M., Deutsch, S., Lebiere, C., et al. (2005, September 26-30, 2005). *Human Performance Models of Pilot Behavior.* Paper presented at the Human Factors and Ergonomics Society 49th Annual Meeting, Orlando, Florida.

Huss, D. G., & Byrne, M. D. (2003). An ACT-R/PM model of the articulatory loop. In F. Detje, D. Dörner & H. Schaub (Eds.), *Fifth International Conference on Cognitive Modeling* (pp. 135-140). Bamberg, GE: Universitas-Verlag Bamberg.

Neri, P. (2010). How inherently noisy is human sensory processing? *Psychonomic Bulletin & Review, 17*(6), 802-808.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review, 115*(1), 101-130.

Salvucci, D. D., & Taatgen, N. A. (2011). *The multitasking mind*. New York: Oxford University Press.

Schooler, L. J., & Anderson, J. R. (1997). The role of process in the rational analysis of memory. *Cognitive Psychology, 32*(3), 219-250.

Sims, C. R., & Gray, W. D. (2004). Episodic versus semantic memory: An exploration of models of memory decay in the serial attention paradigm. In M. C. Lovett, C. D. Schunn, C. Lebiere & P. Munro (Eds.), *6th International Conference on Cognitive Modeling* (pp. 279-284). Pittsburgh, PA: Carnegie Mellon University/University of Pittsburgh.

Taatgen, N. A. (2007). The minimal control principle. In W. D. Gray (Ed.), *Integrated models of cognitive systems*. New York: Oxford University Press.

Wickens, C. D., McCarley, J. S., Alexander, A. L., Thomas, L. C., Ambinder, M., & Zheng, S. (2005). *Attention-Situation Awareness (A-SA) Model of Pilot Error*: University of Illinois at Urbana-Champaign, Institute of Aviation Human Factors Division.