An ACT-R Model of Commercial Jetliner Taxiing

Jeffrey C. Zemla*, Volkan Ustun*, Michael D. Byrne*, Alex Kirlik[†],
Kenyon Riddle[†], and Amy L. Alexander°
*Department of Psychology, Rice University °Aptima, Inc., Woburn, MA
[†]Aviation Human Factors Division, University of Illinois Urbana-Champaign

The Next Generation Air Transportation System (NextGen) seeks to reduce gridlock at airports by, among other things, creating a more efficient surface taxi management system. Addressing this situation creates a difficult evaluation problem; how can new scheduling methods be tested? Present methods generally involve either expensive human-in-the-loop experiments or computer simulations that do not adequately represent the human component of system performance. We have developed an ACT-R model of commercial jetliner taxiing with the ultimate goal of aiding in both of these efforts. The X-Plane commercial flight simulation package provides an environment in which the model can act. That environment is populated with aircraft driven by recordings taken of real aircraft at Dallas-Fort Worth airport, which contain the actual positions of all aircraft on the taxi surface for a given time slice. This also provides us with a rich source of data for model validation, as the model can "replace" one actual aircraft, allowing comparisons between model-generated and pilot-generated trajectories.

INTRODUCTION

As the volume of commercial air traffic has increased, large airports have strained to accommodate increased levels of congestion. One reason for this difficulty arises from the inefficiency of surface traffic management. The task of optimizing the timing and route of each plane from the gate to the runway is computationally difficult, and ground controllers do not have the proper resources to do such optimization. This task becomes even more complex as the amount of surface traffic increases, which leads to delays that cost airlines time, fuel, and money (FAA, 2010) as well as inconvenience (or worse) for passengers.

In recent years, this task has been made easier with the introduction of Airport Surface Detection Equipment – Model X (ASDE-X). ASDE-X is a key component of NextGen (FAA, 2010), which aims to reduce gridlock at airports around the country. ASDE-X synthesizes several positioning systems, such as Automatic Dependent Surveillance-Broadcast (ADS-B), surface movement radar, and multilateration sensors into a more robust set of positioning data. Each of these techniques uses a different technology to locate aircraft on the taxiways. For instance, ADS-B relies on satellite navigation systems (such as GPS), where as multilateration determines an aircraft's position by measuring the time taken for a signal to travel from a plane to multiple receivers within the airport. ASDE-X is able to weigh the strengths and weaknesses of each in order to provide controllers with precise, real-time positions of all nearby planes. This helps ground controllers perform their job more efficiently.

With advancements in surface management technology, researchers have began experimenting with computer algorithms that calculate the optimal sequencing and routing of planes as they move about the taxi surface (Malik, Gupta, & Jung, 2010). However, the current methods for testing these algorithms are limited in several ways.

One common method is to employ human-in-the-loop (HITL) simulations. These simulations involve people coming into a lab in order to pilot flight simulators while ground controllers provide them with instructions. This method of testing is useful because the capabilities of real pilots play an important role in determining the validity of the algorithm. For instance, an algorithm may predict high efficiency by closely spacing planes together, but human pilots may not be able to safely implement these procedures. In addition, the reaction times of pilots can add latency to the system that is not apparent otherwise. While HITL testing can provide realistic results, it suffers from two major drawbacks. Firstly, it can be expensive, totaling thousands of man-hours in order to test new changes. Secondly, HITL simulations do not always account for behavior on a large scale. For instance, if one wanted to predict the rate of runway incursions that arise from several nearby airports over the span of a few months, HITL testing is simply not feasible.

Another common method for testing these algorithms is to use computer simulations, such as the Surface Operations Simulator and Scheduler (SOS^2; Wood, Kistler, Rathinam, & Jung, 2009). Computer simulations overcome the major concerns of HITL testing: they are both fast and comparatively inexpensive. However, current computer simulations have their own limitations. SOS^2 does not dynamically simulate human pilot behavior. Responses to ground controllers are predetermined, meaning that the planes in these simulations always react to air traffic controllers without error and in zero time. Furthermore, off-nominal situations are neither detected nor corrected in these simulations due to the lack of consideration for pilot cognition. While such omissions are not uncommon in the early stages of research on a problem, they expose a serious gap in our ability to accurately predict the outcome of changes to the surface management system.

In this paper, we present a computational cognitive model of pilot taxiing. A cognitive model has the benefits of being both fast and inexpensive, while also integrating key components of human cognition and behavior that may affect the simulations, such as pilot errors and response times.

ACT-R

We constructed our cognitive model using ACT-R 6.0 (Anderson, 2007), a computational cognitive architecture that is capable of simulating human cognition through an interaction of lower-level psychological processes, such as memory retrieval and visual attention. The mechanisms that underlie these lower-level processes have been well vetted in the psychological literature to ensure predictive accuracy. ACT-R is an ideal candidate for modeling pilot taxiing behavior for several reasons. Firstly, it has proven capable of modeling complex tasks in both aviation (Byrne & Kirlik, 2005) and driving (Salvucci, 2006) domains. Secondly, ACT-R is capable of interacting with an external environment in real-time, which allows us to watch and evaluate the model performance in the same way we might evaluate human performance.

ACT-R models are created by specifying the domain-specific procedural and declarative knowledge of the human being modeled. This information is derived from subject matter experts as well as airline procedural documentation. Propositional symbols (or chunks, in ACT-R parlance) are used to denote items in declarative memory, as well as objects in the environment. Procedural memory is represented in the form of production rules. Production rules are IF-THEN rules that require specific conditions to be met (IF) in order to execute a set of actions (THEN). For instance, one production rule might state, “IF there is a hold sign present, and the speedometer indicates a speed greater than 0, and the brake is not being pressed THEN step on the brake.” The behavior resulting from a production rule can take a wide variety of forms, including changes to internal mental states, movements of visual attention, or motor movements. The end result of a run of an ACT-R model is a time stamped list of actions performed by the model called the trace.

X-PLANE

X-Plane 9, a commercial flight simulator, acts as the external environment for our model. Information presented in X-Plane, such as taxiway signage and cockpit displays, is encoded as chunks in the model’s visual field. These chunks can determine which production rule should fire at any given time. In turn, a production rule may produce observable behavior within X-Plane. For instance, if a production rule determines that the throttle should be increased, the throttle in X-Plane is actually increased. Thus, the ACT-R model and X-Plane are dynamically linked.

X-Plane handles the physics necessary to make our simulation realistic. For instance, when our model decides to increase the thrust of the plane, X-Plane determines the acceleration and velocity depending on the type of plane the model is currently piloting. In addition, X-Plane provides detailed maps of airports worldwide, including signage on the taxiways. This enables us to simulate real clearances at real airports, which produces concrete predictions about how well these systems work at any particular airport.

The model communicates with X-Plane using a plug-in infrastructure, which allows our model to read state variables, such as position and velocity. However, since ACT-R does not contain a machine vision component, visual aspects that are crucial to the model’s performance must be redrawn on a proxy interface in a manner that our model can “see.” This proxy window takes the form of a Lisp window, with visual objects marked up such that they can be encoded by ACT-R’s visual system.

The resulting system runs on two machines, a PC running X-Plane and a Macintosh running Lisp and maintaining both the virtual cockpit and ACT-R. The system is depicted in Figure 1.

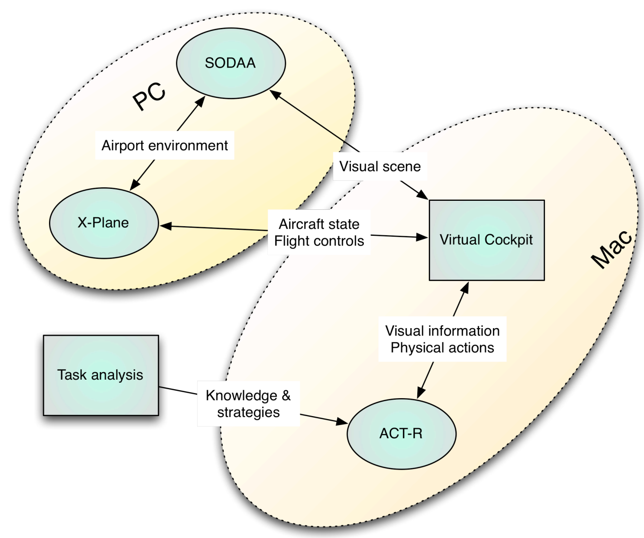

Figure 1. ACT-R communicates directly with the virtual cockpit, both of which run on one machine. In turn, the virtual cockpit communicates with X-Plane, which runs on a separate machine. X-Plane is also able to replay pre-recorded routes, referred to here as the SODAA data, which are used to validate the model.

MODEL OVERVIEW

Prior to constructing the model, we surveyed airline procedural documentation and questioned pilots in order to determine what domain-specific information was necessary to create the model. With this information, we conducted a task analysis that defined the sequence of operations a pilot must perform to taxi a plane. Ultimately, each of the operations in the task analysis was translated into a series of production rules.

The task analysis identified several key components that are required for a pilot to successfully taxi a commercial jetliner. These components include navigating the taxiways, steering the plane, maintaining the speed of the aircraft, and scanning the taxiway for incursions. Each of these components represents a high-level goal that the pilot is responsible for. The details of each component are described in the sections below. There are, of course, additional responsibilities of the pilot that are not accounted for by these four components.

Notably absent are goals for processing incoming and outgoing audio transmissions to air traffic control, as well as a variety of pre-flight items (including checklists). These tasks are absent primarily for tractability, however we hope to integrate aspects of these tasks in later versions of the model.

At the top level, one of the four primary goals is chosen probabilistically. When a goal is completed, it returns to the top-level and begins the process again. The interaction of these goals produces simulated pilot behavior and cognition.

Navigating

The navigation goal provides the model with situational awareness. The model keeps a chunk in memory that maintains the current location (taxiway) of the model, the next taxiway in the clearance instructions, and the action to perform at that taxiway (e.g., hold, turn right, turn left). In order to navigate, the model begins scanning the visual scene for signs located on or near the taxiways. When the model reads a sign, the content of the sign is compared to the navigational chunk stored in memory, and decides what action is appropriate (if any).

Upon seeing a sign indicating the current taxiway, the model checks the navigational chunk to determine if the plane is on the correct taxiway. If this is the case, no action is taken. If the plane is on the wrong taxiway, however, the model must take corrective action, such as radioing ground control, coming to an immediate stop, or attempting to find its way back on track. The current version of the model does not presently cover this, however, this incongruence is documented in the ACT-R trace, so that Monte Carlo simulations can predict how often this type of error occurs.

Upon seeing a sign designating a crossing taxiway, the model checks to see if its content corresponds to the upcoming taxiway listed in the navigational chunk. If it does, the model must then look at the action listed in the navigational chunk to decide what to do next. If it does not, no action is taken.

If the plane is to come to a hold, the model sets the target speed to zero. The actual process of decreasing the throttle and hitting the brake is taken care of by the maintain-speed goal.

If the plane is to perform a turn, the model begins looking at the intersection to determine the distance to the turn. When the plane reaches a critical distance to the intersection, the turning sub-goal (described in the next section) is initiated.

Steering

The model has two distinct steering mechanisms. One mechanism is used only for intermittent corrective steering, while the other is specialized for turning.

Corrective steering. This goal is responsible for small steering adjustments, which are necessary to drive straight down a taxiway. Essentially, the purpose of this goal is to minimize the distance of the plane to the centerline of the taxiway. This involves small-angle corrections and can be modeled similarly to how Salvucci's (2006) model handles highway steering of an automobile (though obviously the physics are substantially different).

Turning. This goal is invoked only when the navigation goal signals that a turn is imminent. Steering a commercial jetliner through a turn is a complex perceptual-motor operation, one for which ACT-R did not contain adequate motor capabilities. Based on data from the Surface Operations Data Analysis and Adaptation (SODAA) tool (Brinton, Lindsey, & Graham, 2010), we had access to the turn trajectories of multiple commercial jetliners, and were able to fit those data using a series of motor adjustments based on the speed of the plane and the approximate distance to the hypothetical point where the turn is expected to be completed. The expected heading of the plane can then be calculated as a function of the tangent line at different points on this curve and the model then adjusts the yoke accordingly to match the new heading value. When the yoke adjustments become sufficiently small, the plane is stable and the turn is complete.

Maintaining Speed

The maintain-speed goal controls the speed of the aircraft. The model keeps a chunk in memory that indicates the current target speed of the plane. When this goal is initiated, the model reads the current speed off of the speedometer, and compares this value to the value of the target speed in memory. If the current speed it too high, the model may apply the brakes. This behavior is stochastic, such that the probability of applying the brakes increases as the current speed of the aircraft increases. The throttle is activated in an analogous manner; if the speed of the plane is too low, the model may apply the throttle, and the probability of doing so increases as the current speed of the plane decreases. Typically, the throttle remains in the idle position for the majority of the taxiing.

Scanning the Taxiway

When the scan-taxiway goal is initiated, the model scans the visual environment for possible incursions. Currently, this is limited to other planes present on the taxiway, but this may be expanded to include other possible incursion targets.

The model scans each plane in the visual field that is nearby. If another plane is encountered, the model must decide how to act. If the other plane is in front of the model's plane on the taxiway, the model checks its current speed and the distance to the other plane, and determines whether it is necessary to reduce speed, or come to a halt. If the other plane is not a potential incursion target, no action is taken.

MODEL VALIDATION

For the ACT-R model to be valuable in HITL experiments or computer simulations, it has to be a valid model. Conceptually, the ACT-R model should be on relatively solid ground in terms of validity due to the validation done on the basic components of the architecture and to the extent that the task analysis correctly captures the taxiing task. However, further validation is crucial and we have a unique opportunity in the case of this particular modeling effort.

As noted previously, we have access to data collected using SODAA at Dallas Fort-Worth (DFW) airport that provides an opportunity for operational validation of the model using the historical data (Sargent, 2010). The SODAA tool dynamically records the position of each plane on the taxiways and nearby airspace, thus fully capturing the real life data for the taxiing jetliners. Rather than bringing pilots into a lab to perform the same task as the model, we can use real-world taxiing data to compare to our model's results.

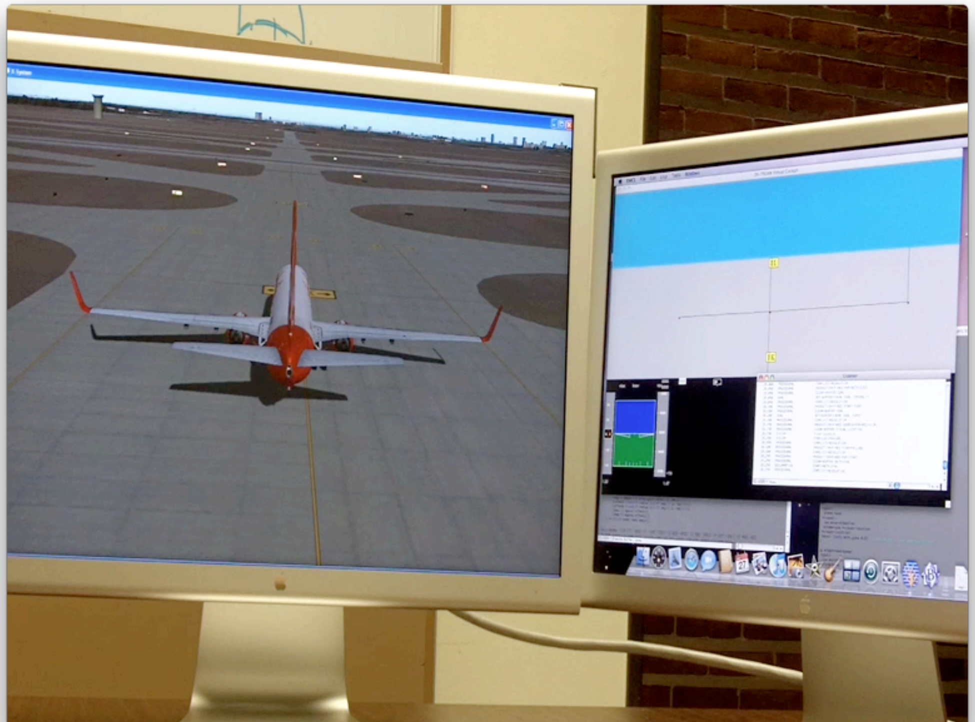

Figure 2. X-Plane is shown on the left monitor, and the virtual cockpit and ACT-R trace are shown on the right monitor.

Thus far, we have only performed face validation as a qualitative assessment of the model's performance by comparing a video of the model performing a specific taxi sequence to a video of the same taxi sequence recorded in the SODAA data in X-Plane. See Figure 2 for a frame of what the running system looks like. We can simultaneously observe the ACT-R model as well as the X-Plane environment that shows the behavior of the controlled aircraft. The model now performs well enough that it is difficult to determine simply from watching the X-Plane view whether it is a replay or whether it is ACT-R in control. This is, in some sense, a form of "Turing test" for the ACT-R model.

However, more quantitative validation is necessary. We are currently in the process of developing the underlying framework that will allow historical data validation. This framework involves letting one jetliner to be controlled by the ACT-R model while all the other jetliners are replays from the SODAA data stream. We can then record the trajectory in both time and space of the jetliner controlled by the ACT-R model and compare it to the data it replaced from the SODAA stream. This will enable a quantitative assessment of our model's performance, though it is not entirely clear exactly what measures or metrics are most appropriate for measuring the degree of deviation between model and data. If the model takes a wrong turn, for instance, that is clearly inappropriate. However, what if the model drives almost identical spatial trajectory, but a few seconds slower or faster than the human pilot? Is that valid enough?

Obviously, there are some open issues with respect to validation. However, unlike other human performance modeling efforts, we are fortunate in that we have a large volume of data against which to validate model performance.

DISCUSSION

The current model has several possible applications. One potential use is to integrate the model with other computational models such as SOS^2 to allow for rapid prototyping of surface taxiing algorithms. Alternatively, the current model may be used to replace humans in HITL simulations. Essentially, the HITL simulations may remain the same as they are now, but instead of having humans driving flight simulators, we can use the ACT-R model to perform the same task. The model may also be useful in providing estimates for human responses times that are not documented in the literature. Thus, if a researcher needs to know how long it takes for a pilot to react to another plane, he or she may develop a distribution of response times derived from a Monte Carlo simulation.

There are also many interesting avenues for extending the model in the future. For instance, audio communication with ground control is likely to be displaced by data link communication in near future. Data link provides a textual transcript of instructions and communications with ground control to the pilot, so that he is able to rely less on his working memory. While this technology is likely to make taxiing safer, the addition of a new cockpit display may influence other aspects of the pilot's task (Byrne et al., 2004). With an ACT-R model, we can predict how this new technology will affect a pilot's ability to perform the task prior to deploying it on a wide scale.

Additionally, the model's decision-making capabilities can be augmented. Byrne and Kirlik (2005) investigated how pilots decide when to make a turn based on time constraints. Following an incorrect clearance can increase the probability of a runway incursion. Though the current version of the model is capable of navigating the taxiways, it overemphasizes the role of working memory in this task and is likely to under predict wrong or missed turns, and provides no guidance once a wrong turn has been made. By augmenting the decision-making capabilities of the model, we can better predictions of runway incursion rates.

Overall, the model has potential implications for the way new surface management systems are designed, tested, and implemented. By providing a fast, inexpensive, and accurate method for simulating traffic management, we can help NextGen achieve its goal.

ACKNOWLEDGMENTS

We would like to thank Ron Carbonari and Jonathan Sivier at the University of Illinois for their assistance with X-Plane integration and the SODAA data.

This research was supported by the National Aeronautics and Space Administration. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of NASA, the U.S. Government, or any other organization.

REFERENCES

Anderson, J.R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Brinton, C., Lindsey, J., & Graham, M. (2010). The Surface Operations Data Analysis and Adaptation tool: Innovations and applications. *Proceedings of the 29th Digital Avionics Systems Conference*.

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., Huss, D.G., Kosorukoff, A., Lin, R., & Fick, C.S. (2004). A closed-loop, ACT-R approach to modeling approach and landing with and without synthetic vision system (SVS) technology. *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting,* 2111-2115. Santa Monica, CA: Human Factors and Ergonomics Society.

Malik, W., Gupta, G., & Jung, Y.C. (2010). Managing departure aircraft release for efficient airport surface operations. *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, Toronto, Canada.

NextGen Integration and Implementation Office, FAA (2010). NextGen implementation plan. Retrieved December 3rd, 2010, from http://www.faa.gov/about/initiatives/nextgen/media/ngip_3-2010.pdf

Salvucci, D.D. (2006). Modeling driving behavior in a cognitive architecture. *Human Factors*, *48*, 362-380.

Sargent R.G. (2010). Verification and Validation of Simulation Models. *Proceedings of the Winter Simulation Conference*, Baltimore, MD.

Wood, Z., Kistler, M., Rathinam, S., & Jung, Y. (2009). A simulator for modeling aircraft surface operations at airports. *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, Chicago, Ill.