A Cross-Model Comparison of Human Performance Modeling Tools Applied to Aviation Safety

Revision 2

Prepared for:

NASA Ames Research Center
Moffett Field, CA 94035

Prepared by:

Ken Leiden
Brad Best, Ph.D.
Micro Analysis & Design, Inc.
4949 Pearl East Circle
Suite 300
Boulder, CO 80301

September 30, 2005

Table of Contents

1 Introduction

The NASA Aviation Safety Program (AvSP) was created to perform research and develop technology to reduce the rate of fatal aircraft accidents in the USA. Under AvSP, the System-Wide Accident Prevention project uses current knowledge about human cognition to develop mitigation strategies to address current trends in aviation accident and incident profiles. System-Wide Accident Prevention is comprised of four elements, one being Human Performance Modeling. The objective of the Human Performance Modeling Element is to develop predictive capabilities to identify likely performance improvements or error vulnerabilities during system operations. During the time period of interest (FY01-FY04), this element investigated the application of human performance modeling (HPM) to predict the performance of flight crews of commercial air transport carriers in two different modeling efforts. The first effort modeled flight crew taxiway operations at Chicago O'Hare airport with an emphasis on predicting taxiway errors (e.g., wrong turns or missed turns). The second effort modeled the use of NASA's synthetic vision system (SVS), which depicts a clear, 3-dimenisional view of terrain and obstacles regardless of the actual visibility or weather conditions. In this effort, the simulated flight crew performed instrument approaches into Santa Barbara airport under instrument meteorological conditions (IMC) under baseline and SVS configurations. In both efforts, HPM proved to be a valuable research tool for understanding these new systems and their impact on human performance.

Five modeling teams were chosen by NASA Ames Research Center to develop human performance models of pilots performing taxiway operations and instrument approaches. Each team was permitted to use their HPM tool of choice to develop their human performance models. The five teams and their respective HPM tools are shown in Table 1 below:

Table 1. Modeling teams and HPM tool of choice

Team affiliation	HPM tool
Rice University/University of Illinois Urbana-Champaign (Rice/UIUC)	Adaptive Control of Thought-Rational (ACT-R) version 5.0
Micro Analysis & Design/ Carnegie Mellon University (MA&D/CMU)	ACT-R using a variant of version 5.0
San Jose State University (SJSU)	Air Man-machine Integration Design and Analysis System (Air MIDAS)
BBN Technologies (BBNT)	Distributed Operator Model Architecture (D-OMAR)
University of Illinois Urbana-Champaign	Attention-Situation Awareness (A-SA)

The modeling teams have documented their modeling efforts in separate reports delivered to NASA in 2004, but currently not available to the public.* Although it would be possible for an individual reader to scrutinize these reports (when they become available) to reveal similarities and differences between the modeling efforts, this would most likely be a daunting endeavor. Realizing this, NASA funded a cross-model comparison task in which the HPM tools and resulting models could be discussed and compared. This paper is the result of that task.

* Reports from 2003 available at http://human-factors.arc.nasa.gov/ihi/hcsl/publications.html under the section "Modeling Approach and Landing with Synthetic Vision Systems - Interim Reports and Publications."

The document is organized as follows: Section 2 gives a brief description of the taxiway operations and instrument approach simulations conducted at NASA facilities. Section 3 of this report discusses the objectives of this cross-model comparison effort. Section 4 presents an overview of the modeling tools to familiarize the reader with each tool's theory of operation and general capabilities. Section 5 is the focal point of the cross-model comparison. This section is further divided into sub-sections to address how specific capabilities such as memory, scheduling, and visual attention are addressed by the modeling teams and their respective tool. Section 6 discusses the usefulness of the HPM tools to the design and evaluation of new aviation technology.

2 Background

Taxiway operations information to support HPM effort

For the taxiway operations modeling effort, NASA provided the HPM teams with information describing pilot procedures during final approach, landing, and taxiway operations. Videotapes were provided of pilots in the NASA Advanced Concept Flight Simulator (ACFS) performing taxiway operations at Chicago O'Hare airport as part of the taxiway navigation and situation awareness (T-NASA) study (Hooey, Foyle, & Andre, 2000). Each trial began on a level approach approximately 12 miles from O'Hare airport. Based on the clearance obtained during final approach, the pilot-not-flying informed the pilot-flying of their location with respect to the runway exit by referring to the airport diagram. After landing (using the aircraft's autoland capability), the aircraft cleared the runway and the flight crew switched to the ground controller frequency. The controller provided a taxi clearance in terms of intersections and directions from the current location to the destination gate. The flight crews were then required to taxi to the gate under limited visibility [runway visual range (RVR) of 1000 ft].

Although the goal of the T-NASA study was to investigate the increased situation awareness and resulting reduction in taxiway errors afforded by the new T-NASA technology, the HPM effort only focused on the baseline trials in which standard flight deck displays were utilized since in these trials abundant taxiway errors occurred (12 major errors in 54 trials). Furthermore, these errors were virtually eliminated with the introduction of the T-NASA display. It was anticipated that the underlying causes of the "baseline" errors could be investigated through HPM. An analysis of the errors was conducted by NASA and provided to the modeling teams (Goodman, 2001) so the modeling teams could focus on errors best represented by their modeling tool capabilities.

Instrument approach information to support HPM effort

For the instrument approach modeling effort, NASA provided a cognitive task analysis of the approach phase of flight (Keller, Leiden & Small, 2003) and human performance data collected from NASA part-task simulations of approaches into Santa Barbara airport (Goodman, Hooey, Foyle & Wilson, 2003). In addition to videotapes of the part-task simulations, NASA collected eye-tracking data that was later post-processed to reveal the number of fixations that occurred on the various flight deck displays and out-the-window view. The modeling teams were given this data to further post-process as needed.

The primary purpose of this effort was to compare pilot behavior using traditional flight deck displays vs. SVS. The traditional displays are the primary flight display (PFD) (see Figure 1) and navigation display (ND) (see Figure 2). The SVS display is depicted in Figure 3. As can be seen by comparing the traditional displays to SVS, SVS not only offers a 3D rendering of the terrain, but includes an overlay of key information from the PFD and ND. How this redundant information would be used was a question for the HPM teams.

For purpose of analysis, the approach was split into four segments to characterize the pilot tasks and information needs[†]:

Segment	Start and End Point	Altitudes	Duration
Segment 1	Start of Trial – Initial Approach Fix	Crossing at 10,000 ft	1.0 min
Segment 2	Initial Approach Fix – Final Approach Fix	10,000 ft – 1,800 ft	7.5 min
Segment 3	Final Approach Fix – Decision Height	1,800 ft – 650 ft	2.5 min
Segment 4	Decision Height – Scenario End	650 ft – 50 ft	1.0 min

These four progressive segments of approach differ not only in duration but in external circumstances and required pilot activities. Within Segment 1, pilots are focused on obtaining an approach clearance from air traffic control (ATC) and setting-up that approach within the auto-flight system. In Segment 2, pilots closely monitor the progress of the approach and configure the aircraft (e.g., set landing gear, adjust flaps and trim). During Segment 3, pilots flying in IMC "break-out" from the cloud ceiling into full visibility and, for all conditions, pilots must visually acquire the runway and confirm proper alignment. By Segment 4, unlimited forward visibility prevails (except in two scenarios where the aircraft never broke out through the cloud ceiling) and pilots must transition to manual control while maintaining proper runway alignment and descent rate.

In addition to the data provided to the modeling teams from NASA, most teams recruited their own subject matter experts (i.e., pilots) to fill in any gaps and gain a further understanding of the pilot task.

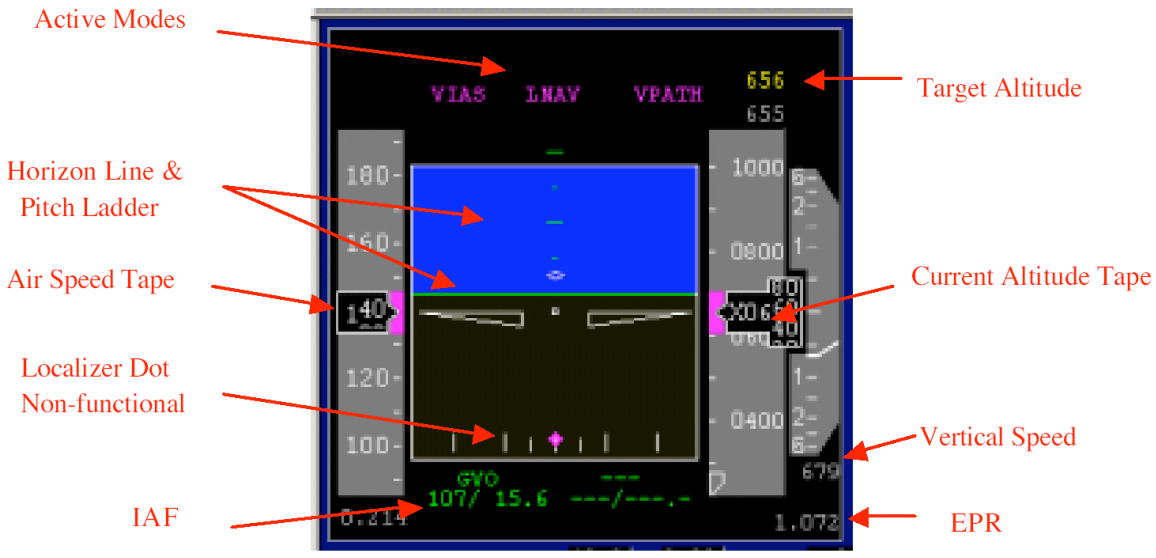

Figure 1. Primary flight display as depicted in the part-task simulations from Goodman, Hooey, Foyle & Wilson (2003).

[†] This description of the segments of approach was copied from Goodman, Hooey, Foyle & Wilson (2003).

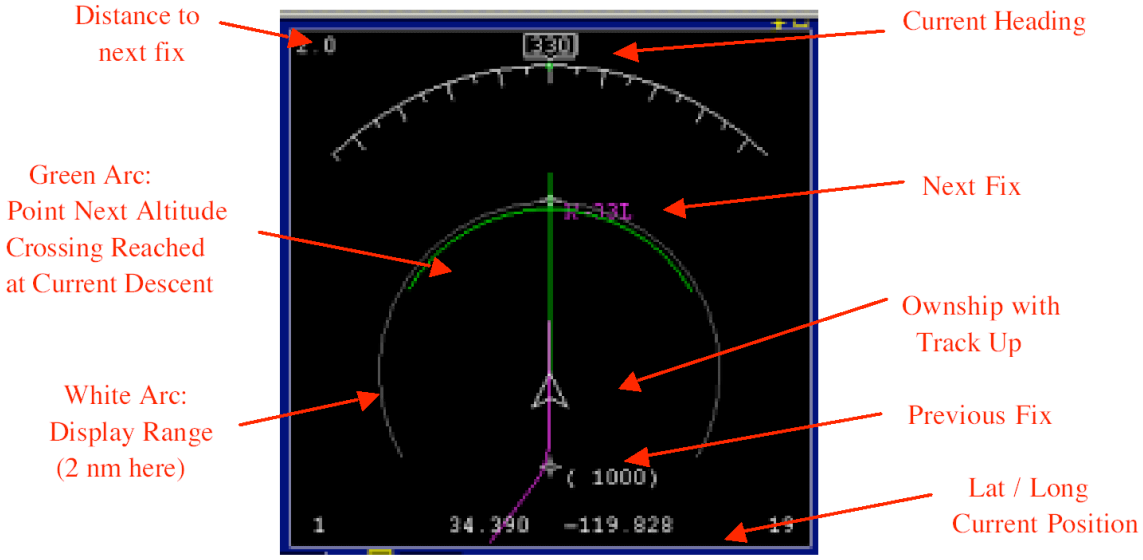

Figure 2. Navigation display as depicted in the part-task simulations from Goodman, Hooey, Foyle & Wilson (2003).

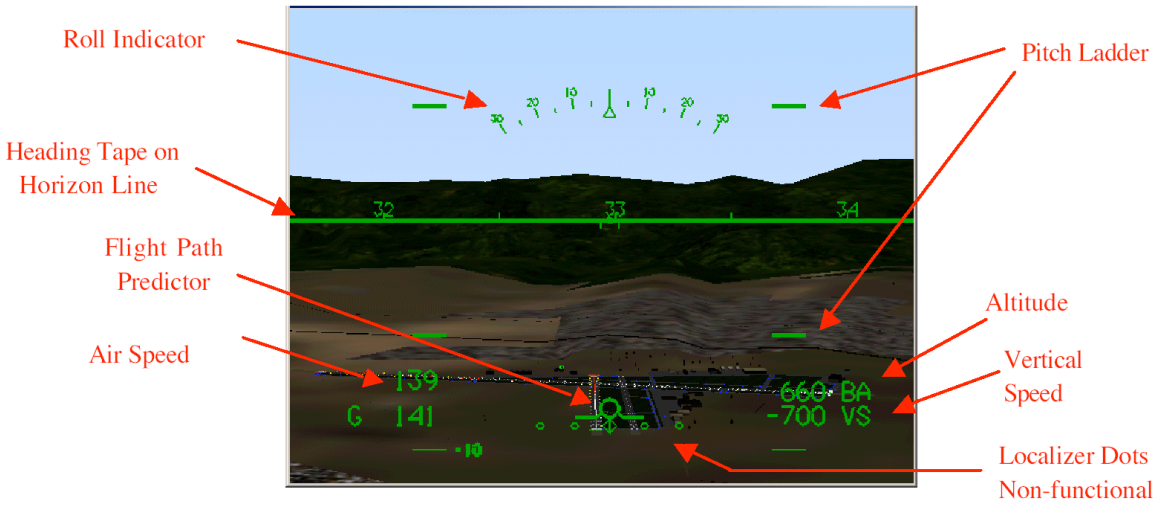

Figure 3. SVS display as depicted in the part-task simulations from Goodman, Hooey, Foyle & Wilson (2003).

3 Objective

The objective is to perform a comparison of the HPM tools presented in Table 1 with an emphasis on the aviation domain. Since some of these HPM tools have been employed in a wide variety of applications and domains, the emphasis on the aviation domain is necessary to keep the comparison manageable while highlighting HPM capabilities that could be beneficial to **aviation researchers** within NASA, other government agencies, and the aviation industry. As mentioned earlier, the HPM Element investigated both taxiway operations and instrument approaches.

However, the comparison presented in this paper primarily emphasizes the latter effort because the modeling teams had more time, funding, and data available for the instrument approach modeling to support an iterative cycle, which resulted in more mature models. Furthermore, the modeling teams had more opportunity to thoroughly document their instrument approach HPM results, which was paramount to this comparison.

Although it is intended that this comparison give NASA and other aviation researchers insight into existing HPM capabilities, future research most likely will require expanding these capabilities. Thus, another objective of this paper is to discuss the usefulness of the HPM tools to the design and evaluation of new aviation technology.

4 Overview of HPM tools

This section gives an overview of each of the HPM tools in terms of available references, theory of operation, general capabilities, functional depiction, programming language, and operating system requirements.

4.1 ACT-R Rice/UIUC

ACT-R is a both a theory and a cognitive modeling tool. The theory describes how humans organize knowledge and produce intelligent behavior. The cognitive modeling tool simulates the application of this theory to the extent that is practically possible. ACT-R researchers consider ACT-R a work in progress and continuously update both the theory and modeling tool to reflect advances in psychology and neuroscience.

ACT-R was originally developed at Carnegie Mellon University under sponsorship from the Office of Naval Research. The first version of ACT-R, version 2.0, was released in 1993. The current version of ACT-R as of the completion of this project is version 5.0, which integrates previous versions of ACT-R with aspects of EPIC's perceptual and motor modules[‡] in addition to other features. The theory underlying version 5.0 is described in Anderson et al (2004) and much of the ACT-R overview presented here is based on that document. Including books, journal articles, and conference proceedings, there are well over 500 ACT-R publications to date, many of which are available at the ACT-R website (http://act.psy.cmu.edu).

ACT-R is open-source and the associated software, models, and tutorials are available from the website. ACT-R is implemented in the common LISP programming language as a collection of LISP functions, data structures and subroutines. ACT-R runs on MacOS, Windows, Linux, and Unix platforms (any platform with a compatible LISP compiler). A number of tools are available for model development and execution, including a graphical user interface to author, run, and debug ACT-R models.

An overview of ACT-R 5.0 is depicted in Figure 4. Each block in the figure represents an independent unit that supports a particular set of cognitive processes as well as the primary region of the brain where those processes are performed. In ACT-R parlance, the three types of units are referred to as modules, buffers, and the central production system. Together, they embody a unified, or integrated, theory of cognition. In concise terms, **modules** serve to represent different types of information such as goals and declarative memory. Each module has an associated **buffer** for holding a single declarative unit of knowledge, referred to as a *chunk*, for two-way exchange with the central production system. The **central production system**, which represents procedural memory (i.e., memory of skill/procedure), detects patterns in these buffers to determine what to do next. In this way, the central production system coordinates between itself and the modules via the buffers.

[‡] see http://www.eecs.umich.edu/~kieras/epic.html

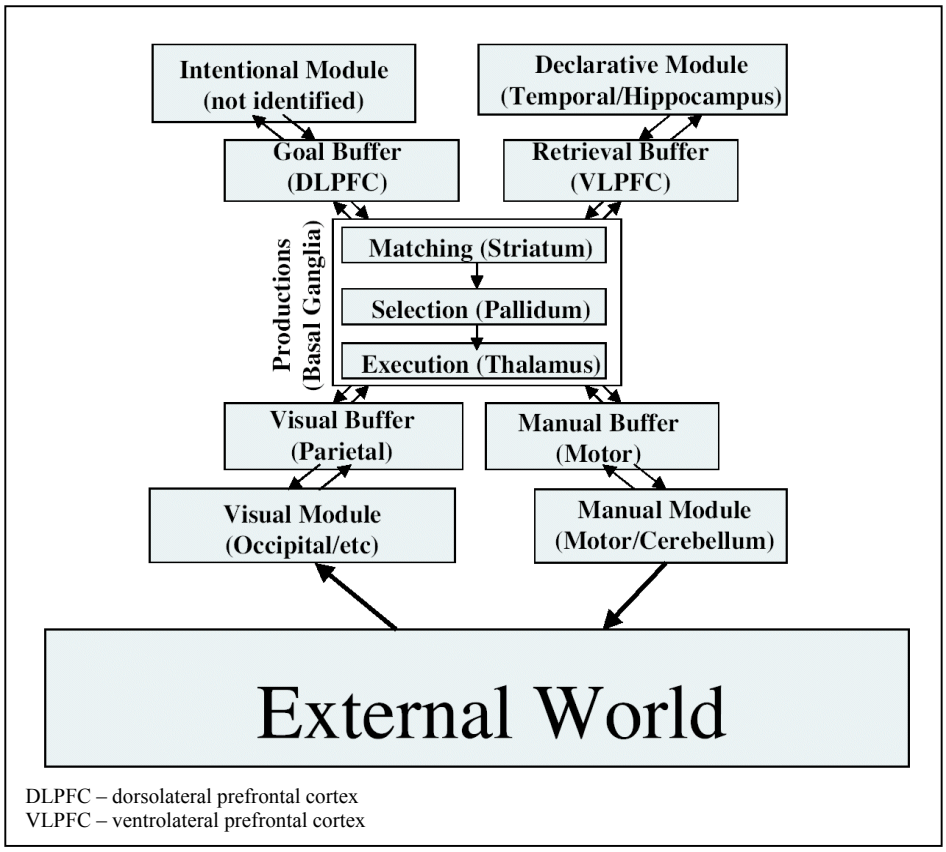

Figure 4. Overview of ACT-R 5.0 architecture[§]

Although not all ACT-R modules and buffers are represented in Figure 4 (e.g., rudimentary modules for speech and audition are not depicted), the ones that are represented emphasize the primary focus of ACT-R 5.0 as follows:

- The declarative module defines the methods for retrieving information from declarative memory. The module determines the probability and latency of chunk retrieval based on past experiences with that chunk and the chunk's relevance to the current context. The retrieval buffer stores the current chunk retrieved from declarative memory.
- The visual module identifies objects in the visual field. There are two visual buffers – one to keep track of object location and one to keep track of object identity, corresponding to the 'what' and 'where' distinctions used by vision scientists to describe human visual processing.
- The manual module and buffer specify how the hands are controlled detailing in particular the sequencing and timing of individual motor movements such as key presses while using an interface.
- The goal buffer keeps track of the current step of the goal. The goal stack from ACT-R 4.0 has been removed and replaced with the existing mechanisms for declarative memory. As a result, goal steps are tracked through the declarative module. The

[§] This figure was provided by Christian Lebiere from Anderson et al, (2004).

intentional module in ACT-R 5.0 as depicted in Figure 4 is currently a placeholder for future refinements.

- The central production system, represented by the matching, selection, and execution blocks in Figure 4, determines the "next step" of procedural memory (i.e., production rule) to execute based on the constraint that only one step at a time can be executed.

4.2 ACT-R MA&D/CMU

The version of ACT-R used by the MA&D/CMU team is not the standard ACT-R 5.0 version, but rather is a variant of ACT-R 5.0 that incorporates particular aspects of ACT-R 4.0. This variant utilizes the ACT-R 5.0 architecture for specifying the interaction of declarative and procedural knowledge and, in general, the integration of multiple buffers for simulating the synchronization of mental activities. However, the variant also incorporates many aspects of the ACT-R 4.0 hierarchical reinforcement scheme, which was based on the goal stack from ACT-R 4.0[**]. This hierarchical scheme enabled a specific type of learning by allowing a top-level goal to take the success and failure of sub-goals into account when determining its utility by propagating success and failure back up the goal stack. This learning mechanism was necessary to the MA&D team's goal of exploring the potential impact of SVS without overly constraining how SVS should be used.

The learning mechanism of interest is the pilot's adaptation to scanning display locations that have rewarding information (information that results in a pilot action) combined with the cost/effort of seeking out that information. Although it is possible to simply specify a scanning pattern, a system that derives the scan patterns from the availability of rewarding information in the environment is more likely to evolve with unknown situations such as the introduction of SVS.

4.3 Air MIDAS

Development of the Air Man-machine Integration Design and Analysis System (Air MIDAS) began in 1985 during the joint Army-NASA Aircrew/Aircraft Integration program to explore the computational representations of human-machine performance to aid crew system designers. Air MIDAS runs on a PC using the Windows 2000 or XP operating system and requires Visual C++ 6.0, Alegro CL, and JAVA software.

MIDAS is a HPM tool specifically designed to assess human-system interaction in the complex, dynamic environments of aviation and air traffic control. Utilizing an integrated approach to human performance modeling, Air MIDAS gives users the ability to model the functional and physical aspects of the operator, the system, and the environment, and to bring these models together in an interactive, event-filled simulation for quantitative and visual analysis. Through its facilities for constructing, running, and analyzing simulations, Air MIDAS can provide measures of operator workload and task performance across a range of conditions. Additional metrics of human-system effectiveness can be derived by users for a given scenario based on observed system or vehicle operating parameters during simulation. Such Air MIDAS output is intended to support the evaluation of current or future equipment, procedures, and operating paradigms.

Operator behavior within Air MIDAS simulation is driven by a set of user inputs specifying operator goals, procedures for achieving those goals, and declarative knowledge appropriate to a given simulation. These asserted knowledge structures interact with and are moderated by embedded models of perception for extracting information from the modeled world and

[**] The goal stack of ACT-R 4.0 was removed from ACT-R 5.0 due to the implausibility of perfect goal memory.

embedded models of cognition for managing resources, memory, and actions (see Figure 5). In this way, MIDAS seeks to capture the perceptual-cognitive cycle of real world operators who work towards their most immediate goals given their perception of the situation and within the limitations of their physical and cognitive capacities. In Air MIDAS, as in the real world, perceived changes in the world – new information or events – may cause changes in the adjudged context of the situation triggering new goals or altering methods to achieve current goals. Such perceived changes may be precipitated through the behavior of other modeled entities or by user specified time-based, condition-based, or probabilistic events set within the simulation (e.g., a system failure or the receipt of an incoming air traffic control clearance). It may, in fact, be the impact of the operator's own actions which lead to changes in context and goals.

This complex interplay between top-down and bottom-up processes makes possible the emergence of unforeseen behaviors. Behavior flows from and is sensitive to the goals, knowledge, and domain expertise imparted to the simulated operator, the perceptual availability of information in the constructed environment, and the nature and timing of external events. This process is partially enabled by micro-models of human performance that feed forward and feed back to the other constituent models in the complex human/system representation. Together, this affords Air MIDAS users flexibility in selecting the focus of study as each of these parameters can be manipulated individually or in concert in order to assess the resulting effects on operator and system performance. Investigative approaches can range from single factor sensitivity analysis to "what if" studies that explore large test matrixes of interacting manipulations and conditions.

While the Air MIDAS framework simulates human behavior at a detailed level, the structure is not intended to be a "unified theory of cognition", perception, or action in the terms specified in ACT-R. The Air MIDAS system should instead be viewed as an attempt at capturing and integrating many of the ‘best practices’ in behavior modeling as expressed through various independent research efforts (e.g., see Card, Moran, and Newell, 1983). That is, the general Air MIDAS research program does not emphasize validating the overall architecture, so it cannot be considered "unified" in the sense of ACT-R, but the individual elements of the system capture many of the elements of human cognitive processes and mechanisms (e.g., modality specific working memories, independent perception and motor processes, attentional processes).

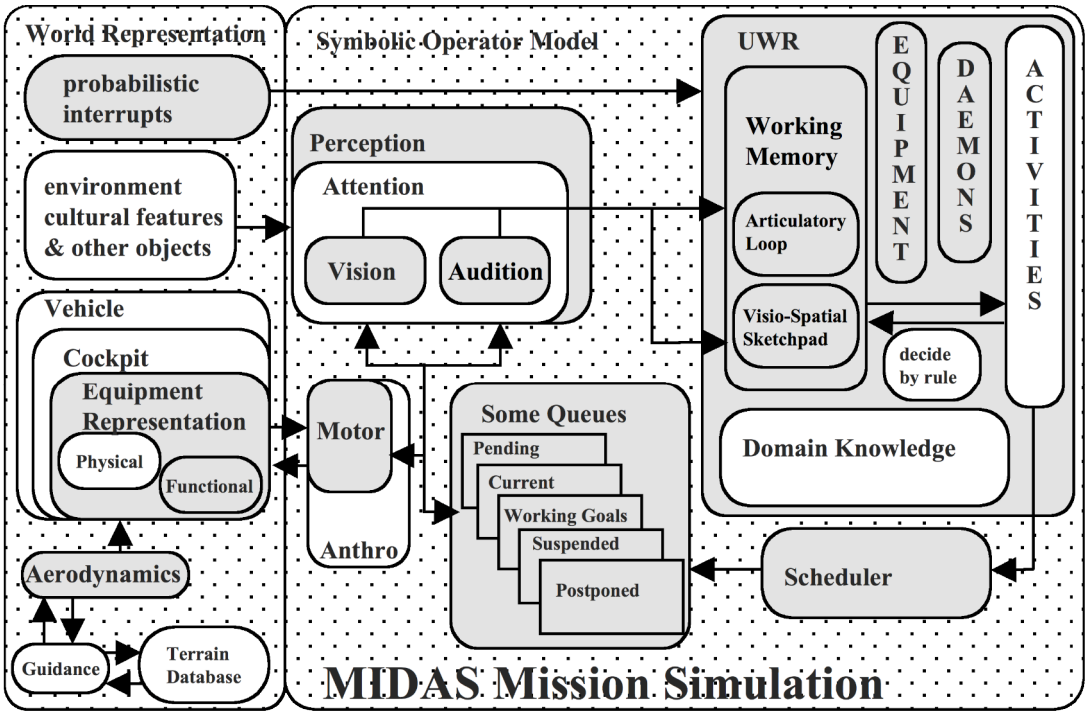

Figure 5 – MIDAS Overview[††]

4.4 D-OMAR

The Distributed Operator Model Architecture (D-OMAR) was developed by BBN Technologies under sponsorship from the Air Force Research Laboratory. D-OMAR is an event-based architecture for developing human performance models or agents for agent-based systems. The agents or human performance models operate in a simulation environment running in real-time or fast-time. D-OMAR supports the notion of an agent whose actions are driven not only by actively seeking to achieve one or more defined goals, but also by reacting to the input and events of the world. The D-OMAR knowledge representation languages were designed to *facilitate the modeling of the human multi-tasking behaviors* of team members interacting with complex equipment in pursuing collaborative enterprises.

The knowledge representation languages that form the basic building blocks for the modeling effort are the:

- Simple Frame Language (SFL) for defining the agents and objects in the simulation world.
- Simulation Core (SCORE) procedure language for defining the behaviors of agents and objects.

The D-OMAR model should run well on most popular hardware/software configurations. It relies on a Lisp implementation known as Franz Lisp and also requires Java. All user interface components of D-OMAR, the graphical editors and browsers, the simulation control panel, and the timeline displays, have been built in Java. D-OMAR is open source software available at

[††] This figure was taken from a chapter by Kevin Corker in a forthcoming FAA-Eurocontrol book on HPM for air traffic management.

http://omar.bbn.com. In addition, the website has D-OMAR references and an online user manual. D-OMAR runs on Windows and Linux (and Solaris in the past). External simulations can communicate with D-OMAR via HLA, CORBA, or a higher performance native-mode D-OMAR communication protocol.

The D-OMAR architecture is not limited to any particular theoretical approach in the design of human performance models. Rather, the flexible architecture enables users to develop models according to their own psychological theory, philosophy, or inclinations. The underlying theory driving the behavior of an agent is whatever the user chooses to implement as long as it is of the perspective that the agent demonstrates both proactive and reactive behaviors. Although D-OMAR does not preclude any theoretical approach, it certainly was developed as a framework to facilitate multi-tasking activities (see Figure 6) in complex environments and consequently it does *not* employ a serial executive controlling process that constrains multi-tasking behavior (e.g., the central production system in ACT-R). Instead, it provides a flexible platform enabling the modeler to specify constraints on parallelism and multi-tasking as appropriate or desirable for the modeling task at hand.

D-OMAR models contention between competing tasks in two ways. The first method simply inhibits or interrupts, depending on priority, parallel tasks that require the same perceptual or motor movement resources (e.g., two tasks that each requires the dominant hand of an operator). The second method models contention and resolution based on established policy (Deutsch, 1998). When contention between tasks is present, policy-based priorities determine the next action. In addition, policy-based decision determines if interrupted tasks are reinitiated from the beginning, resumed from the point of interruption, or suspended altogether. A simple, but clarifying example of policy-driven behavior is communication between controllers and the flight deck. Conversations between individual aircrew members of the flight deck are immediately halted when a radio message from a controller is heard. The flight deck listens to the message, even if it is directed towards another aircraft. Likewise, flight deck-initiated communication must wait until the party-line is clear before the aircrew member can speak.

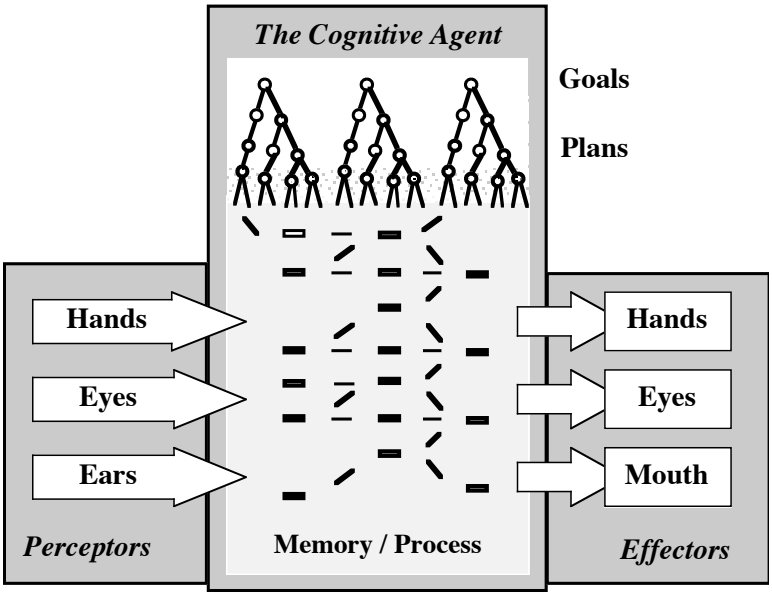

Figure 6 The D-OMAR Cognitive Agent (Deutsch, 1998).

4.5 A-SA

The A-SA model predicts pilot situation awareness (SA) and assessment. A-SA is unlike the other HPM tools discussed in this paper in that A-SA does not attempt to model complete human performance. For example, the model does not incorporate action-related phenomena or low-level cognition. A-SA is typically run stand-alone as a dynamic model, but it could be embedded into a complete HPM modeling frameworks (such as ACT-R, Air MIDAS or D-OMAR) to model errors of situation awareness, such as making a wrong turn on a taxiway. A-SA has thus far been tailored to the aviation domain, but could be applied to other domains provided the analyst is capable of specifying the "knowledge" that comprises SA in the applicable domain.

A-SA is composed of two modules representing attention and belief updating (see Figure 7). The **attention module** describes the allocation of attention to events and flight deck displays within the aircraft environment [approximates Endsley's (1988) stage 1 SA (perception)]. The **belief updating module** describes SA in terms of understanding the current and future state of the aircraft [approximates Endsley's stages 2 & 3 SA (comprehension and projection)].

Attention module

Underlying the allocation of attention module is a static, analytic model known as SEEV (for salience, effort, expectancy and value). SEEV predicts the probability of attending to a particular flight deck display or out-the-window view. SEEV is based on the premises that a:

Pilot's attention is influenced by two factors:

1. Bottom-up capture of salient or conspicuous events (visual or auditory)
2. Top-down selection of valuable event information in expected locations at expected times

Pilot's attention is inhibited by a third factor:

3. Bottom-up level of effort required, which encompasses both the effort of moving attention elsewhere and the effort required due to concurrent cognitive activity

Belief updating module

The module for understanding of the current and future state of the aircraft is based on a belief updating. SA is updated any time new pieces of information are encountered. SA has a value between 0 and 1 with 1 denoting perfect SA. The new pieces of information influence SA based on the value and attentional allotment of those pieces. For example, correct information of high value would improve SA whereas incorrect information of high value would degrade SA. In either case, SA is updated using the new information via an anchoring and adjustment process. When no new information is available, SA is assumed to decay slowly (time constant of 60 s). When irrelevant information is encountered, SA is assumed to decay faster (time constant of 5 s) because of interference with long-term working memory. The value of SA is used to guide subsequent attentional scanning – good SA guides attention towards relevant information whereas poor SA guides attention towards less relevant information. In addition, SA determines the likelihood the pilot will choose correctly when confronted with a decision point (e.g., turn or go straight when encountering a taxiway intersection). If the pilot has poor SA, default behavior provided by the analyst influences the decision (e.g., when in doubt, go straight), which may be erroneous.

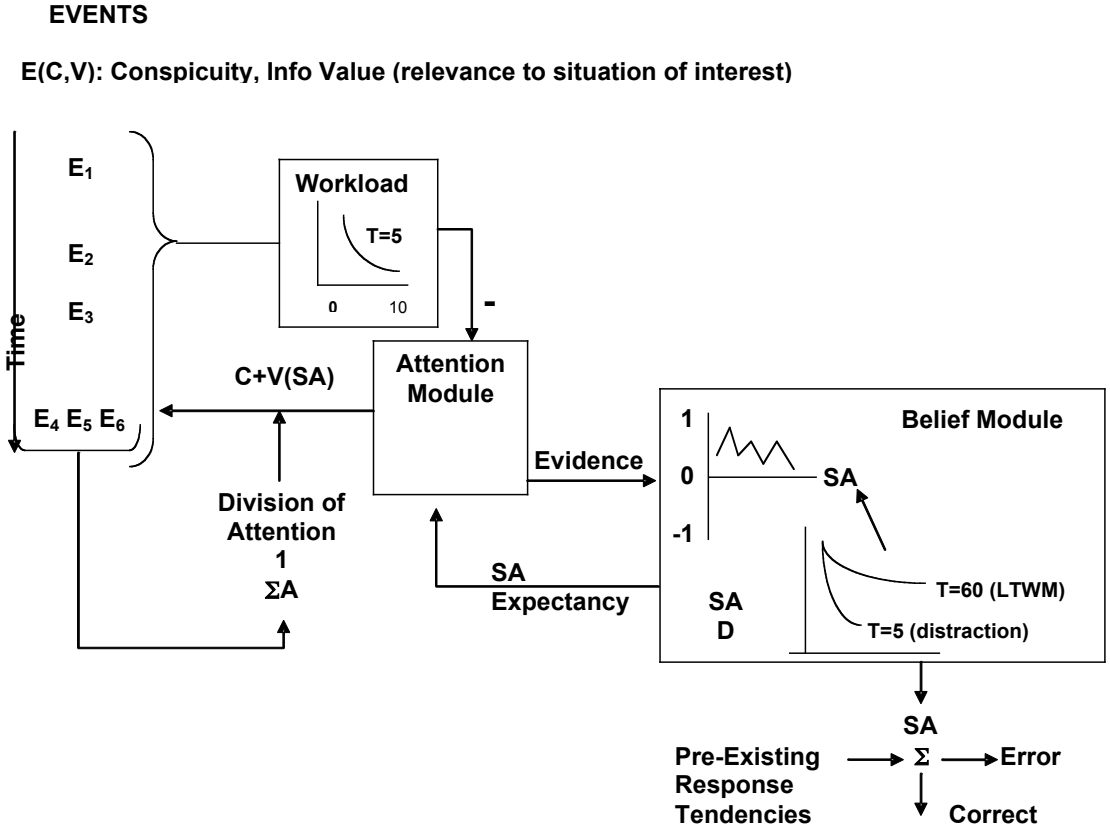

Figure 7. The A-SA model[‡‡]

[‡‡] This figure was taken from the A-SA chapter of the forthcoming NASA-sponsored AvSP HPM book

5 Modeling Capabilities

This section describes the HPM capabilities developed and demonstrated by the five modeling teams under the NASA HPM Element with an emphasis on the instrument approach modeling effort. The descriptions are intended to provide insight into each HPM tool's level of fidelity for a given capability. The capabilities address a range of human performance considerations:

- External environment
- Memory
- Scheduling and multi-tasking
- Crew interactions
- Visual attention
- Workload
- Situation awareness
- Error prediction
- Learning
- Resultant and emergent behavior

5.1 External environment

From an HPM perspective, the external environment is the environment in which the simulated operator(s) interacts. The external environment can represent something simple (e.g., an automated teller machine) or something quite complex (e.g., a nuclear reactor and its control room). In the aviation domain, the external environment typically represents an aircraft's state (position, velocity, attitude) and performance characteristics; dynamic responses to pilot actions; some representation of flight deck displays and instrumentation; and the atmosphere and surroundings through which the aircraft travels.

The level of fidelity at which a complex external environment should be represented is often a difficult decision for the modeler to make. On the one hand, a high-fidelity environment most closely represents the closed-loop behavior between a pilot's actions (or inactions) and the aircraft's response. Thus, if a pilot performs a task poorly, the aircraft flight dynamics should exhibit the appropriate response, which in turn may require a corrective action by the pilot. This feedback mechanism is often a crucial aspect of HPM, particularly with respect to workload. Without it, the modeler risks concealing important human performance issues. On the other hand, a high-fidelity model of the external environment can be difficult, time consuming, and prohibitively expensive to develop. Fortunately, there are commercial-off-the-shelf (COTS) flight simulators readily available to provide a high-fidelity external environment so the issue becomes one of integration rather than development. Not that integration does not have its own issues – the process of passing information between simulations can also be difficult, time consuming, and require specialized expertise. Thus, when considering the fidelity of a complex external environment, the modeler must carefully consider these issues, particularly since the decision is made early in the project and midcourse corrections may not be feasible.

It is important to note that ACT-R, Air MIDAS, and D-OMAR have all implemented High Level Architecture (HLA), developed by the Department of Defense to address simulation interoperability issues, during previous modeling projects. However, unless specifically required by a particular project (which is sometimes the case for military applications) most HPM teams prefer to use other methods for simulation interoperability due to HLA's complexity and overhead.

For the instrument approach HPM effort, a B-757 commercial air transport carrier performing an area navigation (RNAV) approach was assumed. During the RNAV approach, the flight path was determined using lateral and vertical navigation (LNAV, VNAV) modes in which waypoints

(latitude, longitude, and altitude) entered a priori defined the lateral and vertical path of the aircraft. Under this autoflight mode, the aircraft essentially flies itself. Interestingly, the only control inputs needed from the pilots were speed settings via the mode control panel, flap settings, and lowering the landing gear. In addition, there were no significant winds, wind shear, or other off-nominal events that would alter the aircraft's flight path significantly from a nominal path prior to breaking through the cloud ceiling or reaching decision altitude, whichever came first (pilots typically take manual control of the aircraft at this point, but the modeling teams did not attempt to model these motor tasks). These assumptions are stated here to emphasize that there is reasonable justification for representing the aircraft dynamics by simple kinematic equations and scripted events (as was done by the ACT-R MA&D/CMU, D-OMAR, and A-SA teams) without a significant impact on the pilot-action/aircraft-response feedback mechanism. *For this particular modeling effort,* it is the belief of the authors that this lower fidelity modeling of the external environment provides an adequate rendering for the HPM component predictions. That said, the primary drawback to these simpler external environments is that the models' extensibility to future work may be hindered if the future work must demonstrate a closely coupled feedback mechanism between pilot-action/aircraft-response (and vice versa). On the other hand, the ACT-R Rice/UIUC and Air MIDAS teams chose to integrate their HPMs with higher fidelity flight simulators, avoiding this potential drawback. Furthermore, by coupling to a higher fidelity flight simulator, some aspects of model verification become readily apparent. For example, if the speed brakes were not retracted on landing and a tail strike occurred, it should be obvious to the modeler that there is a bug in the model or the pilot experienced some type of distraction during the run that needs investigating. In contrast, it would be unlikely that the lower fidelity models would capture the dynamics of a tail strike (of course, if the modeler had the foresight to include a flag to identify that the speed brake was not retracted, the effect of identifying the problem would be the same).

Table 2. External Environment Synopsis

Modeling tool/team	External environment representation
ACT-R (Rice/UIUC)	X-Plane, a commercially available PC-based flight simulator package
ACT-R (MA&D/CMU)	Simple aircraft kinematic equations using IMPRINT discrete event simulation
Air MIDAS (SJSU)	PC Plane, a NASA-developed PC-based flight simulator
D-OMAR (BBNT)	Simple aircraft kinematic equations internal to D-OMAR
A-SA (UIUC)	Does not require an integrated external environment

ACT-R Rice/UIUC

The ACT-R Rice/UIUC team connected their model to X-Plane, a PC-based flight simulator software package. X-Plane is available for a small fee and includes 40 aircraft types including props and jets, both subsonic and supersonic. X-Plane scenery is world-wide and includes 18,000 airports in its database. The default control for flying an aircraft is the keyboard/mouse/joystick. To work around this, the ACT-R Rice/UIUC team built a 2-way network interface using user datagram protocol (UDP) to allow information to pass between ACT-R and X-Plane such that the simulated pilot is actually flying the aircraft. X-Plane information from the navigation display (ND), primary flight display (PFD), mode control panel (MCP), and out-the-window view is passed to ACT-R so within the ACT-R environment, this information could be accessed and "viewed" by the simulated pilot during a scan. X-Plane information that is numeric is passed

numerically to ACT-R. Information that is based on interpretation such as alignment with a runway or the visibility out-the-window must first be translated into numeric values. The combined ACT-R/X-Plane simulation runs in real-time. Reasonably accurate time synchronization is accomplished implicitly by requiring each simulation to synch to its internal clock.

For the instrument approach modeling effort, the ACT-R Rice/UIUC team estimated that between 40% and 60% of the total model development time was required for modeling the external environment. This is further decomposed into about 10-20% of the time for the communication link between ACT-R and X-Plane and 30-40% to represent the information depicted on the displays.

ACT-R MA&D/CMU

The ACT-R MA&D/CMU team represented the external environment using the Improved Performance Research Integration Tool (IMPRINT). IMPRINT was developed by MA&D through direction of the U.S. Army. Like ACT-R, IMPRINT is itself a human performance modeling tool, but IMPRINT simulates human performance at a larger level of granularity as compared to the cognitive level of ACT-R. In addition, IMPRINT operates in a discrete event simulation framework that facilitates modeling both human operators and the external environment through use of a graphical tool for rapid model generation.

For the instrument approach modeling effort, the delineation between ACT-R and IMPRINT is as follows: ACT-R modeled only the pilot-flying. Information about the external environment was provided by IMPRINT. This included modeling the tasks of the pilot-not-flying, ATC communication, the aircraft dynamics (using relatively simple kinematic equations), and the flight deck displays and controls. With respect to the ND, IMPRINT did not represent "the directional aspects of the flight" (from their SVS final report), which did have an impact on the visual attention allocation as discussed in section 5.5.

The connection between ACT-R and IMPRINT was provided by the Component Object Model (COM) programming interface, which supports a relatively low bandwidth connection. The elements passed across the COM connection from IMPRINT to ACT-R allowed updating the various ACT-R chunks corresponding to the pilot's knowledge and awareness of the displays and controls. Conversely, information flow from ACT-R to IMPRINT consisted of the various actions the ACT-R simulated pilot is capable of performing (e.g., lowering the landing gear, setting flaps).

For the instrument approach modeling effort, the ACT-R MA&D/CMU team estimated that 70% of the total model development time was required for modeling the external environment. This is further decomposed into about 20% of the time for the communication link between ACT-R and IMPRINT and 50% to develop the aircraft dynamic model and represent the aircraft state information on the displays.

Air MIDAS

Similar to the ACT-R Rice/UIUC team, the Air MIDAS team connected their model to a flight simulator software package. In this case, the Air MIDAS team selected PC Plane, a NASA-developed PC-based flight simulator. Dynamic link library (DLL) functions generate aircraft inputs and provide time synchronization. Microsoft Access, a database application, serves as middleware between Air MIDAS and PC Plane to hold numeric information representing the ND, PFD, MCP, and OTW view. One notable difference between the Air MIDAS team's approach and ACT-R Rice/UIUC team's approach is that the Air MIDAS approach provides more precise

time synchronization, since both PC Plane and Air MIDAS are controlled by an independent scheduler (i.e., the DLL functions).

For the instrument approach modeling effort, the Air MIDAS team estimated that 30% of the total model development time was required for modeling the external environment. This includes the PC Plane time control mechanism, model of displays, and the communication link between Air MIDAS and PC Plane.

D-OMAR

The external environment for D-OMAR was D-OMAR itself. D-OMAR simulated the aircraft dynamics (using relatively simple kinematic equations), the flight deck displays and controls, runway configuration, and ATC communication. The external environment developed for the taxiway operations modeling effort was leveraged extensively for the instrument approach modeling effort. The D-OMAR team estimated that less than 25% of the total model development time was required for modeling the external environment. As one would expect, this is a smaller percentage of time than the ACT-R and Air MIDAS modeling teams because internal integration is typically more efficient than integration with another simulation.

A-SA

The SEEV model of A-SA requires coefficient estimates for effort, bandwidth (frequency that data is changing on a display), relevance, and task priority. As such, the external environment needs to be represented only to the extent that reasonable estimates of these coefficients are possible. Hence, a detailed representation of the aircraft's state as a function of time is **not** required. Two examples can clarify this point. First, the effort coefficient is dependent on how the flight deck displays are situated with respect to each other (as can be depicted by a simple mock-up of the flight deck) where transitions between vertically-arranged displays require more effort than horizontally-arranged displays. Second, the bandwidth coefficient can be estimated (and held constant) for a segment of flight in which the bandwidth of a key parameter on the display remains nearly constant. Alternatively, for a more accurate estimate, the bandwidth coefficient can be derived from the actual bandwidth of parameters extracted from flight simulator data without requiring direct integration between A-SA and the flight simulator.

For the other component of A-SA, namely the belief updating module, discrete events and the subjective weights for the salience and relevance of those events contribute to changes in SA. The external environment need only be represented to the extent that the discrete events are adequately captured, which is possible by developing a scripted event timeline from flight simulator data or other representative sources.

5.2 Memory

Memory decay is a frequently discussed subject of human performance modeling. This section discusses the assumptions behind the simulation of memory for each of the modeling teams.

Table 3. Memory Modeling Synopsis

Modeling tool/team	Memory representation
ACT-R (Rice/UIUC)	Sophisticated model for declarative memory including terms for base-level activation and spreading activation
ACT-R (MA&D/CMU)	Similar to above, but includes terms for partial matching instead of spreading activation
Air MIDAS (SJSU)	Working memory capacity and the time duration that memory chunks remain in working memory – the functions associated with interference and decay – are user-specified parameters.
D-OMAR (BBNT)	Assumes perfect memory
A-SA (UIUC)	A-SA does not attempt to represent individual chunks of memory. However, overall SA is assumed to decay with time

ACT-R Rice/UIUC and ACT-R MA&D/CMU

Two types of memory, declarative memory and procedural memory, form the foundation of ACT-R. Declarative memory stores facts and events (i.e., chunks) whereas procedural memory stores associations between stimuli and response that indicate how to do something without consciously thinking about it. From a report-organization and practical perspective, procedural memory is more closely associated with the scheduling and learning sections of this report, and thus is discussed in more detail in those respective sections (5.3 and 5.9).

Before a more thorough description of declarative memory is presented, it is worth noting that ACT-R does not explicitly make a distinction between declarative memory and working memory as do some of the other modeling tools. In ACT-R, working memory is simply a subset of declarative memory that is active (or highly activated in ACT-R parlance). As discussed in detail below, if a chunk is not activated sufficiently then it would not be retrieved, but the activation is dependent upon other factors beyond short-term decay.

In ACT-R, a chunk of declarative memory is defined by both its type (i.e., category of knowledge), its slots (i.e., category attributes), and their values. For example, a type of chunk called "addition-fact" would have slots/attributes for the two operands and the sum (e.g., "2 + 3 = 5"). The chunk type is used in the goal and retrieval buffers to explicitly filter chunk retrievals to the relevant subset of the all chunks residing in declarative memory. After this filter is applied, a critical element of ACT-R's theory is the "activation" process for retrieving chunks – the higher the chunk's activation, the higher the probability that the chunk would be retrieved and, if retrieved, the faster the chunk's retrieval.

Activation (i.e., the value of chunk) is the sum of the following components:

- Chunk's usefulness to past experiences, referred to as **base-level activation**
- Chunk's relevance to the current context, referred to as **spreading activation**
- Chunk's similarity to other chunks, referred to as **partial matching activation**

- A noise term

These components are discussed in more detail because the two ACT-R modeling teams had different approaches to incorporating them into their respective models.

Base-level activation

Base-level activation is the most successfully and frequently employed component of the ACT-R theory. The base-level activation of a chunk grows with frequency of use and decays with time. In ACT-R, there are two formulas for specifying the base-level activation B of chunk i. The first is the more computationally intensive of the two and is represented by the following equation:

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \qquad \text{Base-level learning equation}$$

where n is the number of times the chunk has been accessed, t_j is the time since the j^{th} practice, and the default value for decay parameter d of 0.5 has been determined empirically to be useful across a range of applications.

The computational more efficient, but less accurate equation is:

$$B_i = \ln\left(\frac{n}{1-d}\right) - d\ln(L) \qquad \text{Optimized learning equation}$$

where L is the elapsed time since the creation of chunk i.

For the instrument approach modeling effort, the ACT-R Rice/UIUC team employed a hybrid learning equation, which allowed tracking of just the most recent references of a chunk using the "full" base-level learning equation and then using the optimized learning equation for the older references as a compromise between speed and accuracy. Applied to a specific domain such as aviation, one can presume from the above equations that populating the base-level parameter database can be a time consuming and imprecise effort particularly if emulating realistic pilot experience. Instead, the ACT-R Rice/UIUC team arbitrarily populated the initial database (e.g., the mapping of flap settings to speed) to correspond to very high values of base-level activation so that the required chunk would be successfully retrieved.

In contrast, the ACT-R MA&D/CMU team completely disabled base-level learning. In its place, they set the base-level activations of the knowledge chunks to a constant value that they believed was high, but reasonable. Their reasoning was that knowledge pertaining to the pilot task (e.g., how to set each control based on the values of various state variables) was so well practiced that it should be retrieved quickly and reliably such that the usefulness of the chunk to past experiences would remain constant over the course of the simulation.

Spreading activation

Generally speaking, spreading activation refers to how a chunk of memory receives an activation signal from a related chunk and then in turn passes along the activation signal to each of its direct neighbors (i.e., directly connected nodes in a semantic network). In ACT-R, spreading activation refers to how the chunk of memory in the goal buffer activates (or "excites") chunks that have a direct connection to that chunk. (Note that unlike the general theory of spreading activation, secondary or indirect chunks are not explicitly activated in ACT-R 5.0.) The equation for spreading activation to a chunk in declarative memory is:

$$\textit{Spreading activation to chunk } i = \sum_{j=1}^{n} W_j S_{ji}$$

where j represents each element in the goal buffer and is referred to as a source of activation, n is the number of sources or elements, W_j is the weight of the source, and S_{ji} is the strength of the association between a source in the goal buffer and a chunk in declarative memory.

This equation is most easily understood by example. Assume the goal is to retrieve "small brown animal" from declarative memory (and one assumes a list of chunks of type "animal" exists in declarative memory with attributes of "size" and "color"). In this case, "small" and "brown" are sources of activation in the goal buffer. Any fact/chunk of type "animal" in declarative memory that is encoded with "small" (such as squirrel or raccoon) would receive activation from the above equation. Likewise, any chunk that contains "brown" (such as squirrel or bear) would also receive activation. Using ACT-R defaults, squirrel would be the chunk retrieved because it receives activation from two sources (both "small" and "brown") whereas the other chunks (raccoon and bear) only receive activation from one source (either "small" or "brown"). If there is more than one chunk that matches (e.g., if both "squirrel" and "prairie dog" are stored in declarative memory), then the one with higher base-level activation would be retrieved (assuming no activation noise).

By default, ACT-R weights each source in the goal buffer equally (typically $W_j = 1/n$), which makes sense in the above example since both "small" and "brown" are equally important in the retrieval process. In contrast, how to represent the strength of association between a source in the goal buffer and a fact/chunk in declarative memory is not so obvious. The default method is:

$$S_{ji} = S - \ln(fan_j)$$

where S is typically set to a value around 2 and fan_j is the number of facts/chunks in declarative memory associated with source j in the goal buffer plus one (for source j being associated with itself). Thus, as the number of associations increase, the strength of association decreases. If the modeler has evidence to suggest that the strength of association is not adequately represented by the default method, specific strengths can be input into the model instead.

From an experimental psychology perspective, spreading activation in ACT-R is feasible to investigate because an experiment can be designed specifically to limit the number of associations between sources and facts to manageable amounts. However, from a domain-specific perspective such as aviation, spreading activation is less useful, not so much for what is modeled, but rather for what is *not* modeled. In other words, the ACT-R modeler can readily incorporate the minimum associations between sources and the corresponding facts, but it would be extremely difficult if not impossible to incorporate all, or even most, of the other associations that exist based on a pilot's operational experience. In terms of ACT-R modeling performance, the impact of this would most likely be that the time to retrieve information is faster than actually occurs in real pilots. (Recall that for the instrument approach modeling effort, the base-level activation was set to a high value by both teams so this effect is probably not noticeable.)

For the instrument approach modeling effort, the MA&D/CMU disabled the spreading activation component for the above reason, but more importantly they believed that the partial matching activation was a bigger driver (see below) because the aviation-specific chunks are distinguished by similarities rather than associative strengths. On the other hand, the ACT-R Rice/UIUC team employed spreading activation using the minimum associations and ACT-R default values.

Partial matching activation

Partial matching activation addresses the similarity between chunks in declarative memory and is the mechanism responsible for simulating erroneous retrievals such as memory slips and lapses. In ACT-R, a declarative memory retrieval request specifies both the type of chunk being requested and the attributes/slots of that chunk. The partial matching component addresses the

similarity between an attribute/slot in the retrieval request specification and chunk(s) in declarative memory. It is represented by the following equation:

$$Partial\ matching\ activation\ to\ chunk\ i = \sum_{k=1}^{n} P M_{ki}$$

where k represents the value of an attribute of the chunk in the retrieval request; n is the total number of attributes in the retrieval request, M_{ki} is the similarity between the value k in the retrieval request and the value in the corresponding attribute of chunk i in declarative memory, and P is the weight of the similarity. The MA&D/CMU defined similarity of M_{ki} over the [0,1] interval with 0 meaning complete dissimilarity and 1 meaning perfect similarity. They used partial matching only for numerical values, which provided the possibility for erroneously retrieving nearby numbers (in the ordinal sense) from memory. They assumed the similarity between two numbers k and i to be based on *ratio similarity*:

$$M_{ki} = \frac{\min(k,i)}{\max(k,i)}$$

Ratio similarity has the advantage of scaling well across many different levels in a parameter-free fashion. (In the ACT-R world, the term "parameter-free" refers to not having to tweak standard input parameters to achieve acceptable performance). The MA&D/CMU team used it for all numbers in their model. They assumed a constant value of 5 for P.

Lastly, it is worth mentioning that the Rice/UIUC team chose to disable partial matching.

Retrieval latency

With base-level, partial matching, and spreading activation discussed above, the complete equation for the activation A of chunk i is represented by the following equation:

$$A_i = B_i + \sum_{j=1}^{n} W_j S_{ji} + \sum_{k=1}^{n} P_k M_{ki} + \varepsilon$$

where _ is the only parameter not yet described and represents a noise term.

Once A_i is calculated, the probability and latency of retrieval can be determined. If the activation of a chunk is greater than a retrieval threshold value τ, then the probability of retrieval P of chunk i is specified by the following equation:

$$P_i = \frac{1}{1 + e^{-(A_i - \tau)/s}}$$

where s controls the noise in activation levels and is typically set to 0.4.

When a retrieval request is made, the amount of time T it takes until the chunk i is retrieved and available in the retrieval buffer is:

$$T_i = Fe^{-A_i}$$

where F is the latency factor. (Note that τ and F vary substantially from model to model, but have been shown to be related by the approximation: $F \approx .35e^{\tau}$). If no chunk matches the retrieval request or no chunk has an activation which is greater than the retrieval threshold τ then a retrieval failure will occur.

The time it takes for the failure to be registered in the retrieval buffer is:

$$T_i = Fe^{-\tau}$$

Air MIDAS

The memory structure implemented in Air MIDAS is referred to as the updateable world representation (UWR) (see Figure 5) and models both long-term memory and working memory. Long-term memory is represented by a semantic network and embodies the declarative knowledge of the pilots. The time to retrieve a chunk from long-term memory is assumed to be constant and specified by the fetch time parameter.

Working memory is the subset of UWR that can be interfered with, forgotten, and replaced. A chunk in working memory becomes active when it is referenced by a task/activity or when it is updated by perception. A chunk leaves working memory when the activity that references it is completed, working memory capacity is reached, or duration of time is exceeded. In terms of implementation, working memory chunks are subject to decay by default. (The modeler can override the default settings such that individual chunks do not decay). Activation level for a chunk is determined by the decay rate on each simulation time step. When a chunk's activation level falls below a retrieval threshold the chunk is marked un-retrievable and is no longer available for information retrieval. Working memory capacity and the time duration that chunks remain in working memory – the functions associated with interference and decay – are user-specified parameters. Air MIDAS assumes the "model human processor" of Card, Moran and Newell (1983) for these parameters. Decay rates are assumed to be linear and a single retrieval threshold applies to all working memory chunks. Lastly, Air MIDAS has standard values that specify how long it takes to store visual and auditory stimuli in working memory.

D-OMAR

For the instrument approach modeling effort the D-OMAR team assumed perfect memory and the time to retrieve an item from memory was instantaneous.

A-SA

The memory component of A-SA is straightforward and does not attempt to represent individual chunks of memory. Recall that A-SA predicts a value of SA between 0 and 1 with 1 representing perfect SA. The memory component focuses on the decay of SA over time. Qualitatively, it represents the notion that preservation of SA is a resource-demanding process requiring rehearsal and is not likely to be continuous, particularly when events irrelevant to SA are encountered. SA is assumed to decay faster (time constant of 5 s) when irrelevant information is encountered because of interference with this rehearsal. SA is assumed to decay more slowly (time constant of 60 s) when no new information is available, representing uninterrupted decay of long-term working memory.

5.3 Scheduling and multi-tasking

This section discusses the modeling representation of pilot task scheduling as determined by task priorities, human resource limitations, and system resource constraints.

Table 4. Scheduling and Multi-tasking Synopsis

Modeling tool/team	Scheduling and multi-tasking representation
ACT-R (Rice/UIUC)	Scheduling is a stochastic process performed by production rules. In practical terms, the production rules encapsulate both basic and domain-specific human behavior. Multi-tasking was not implemented for this modeling effort (the ACT-R architecture does permit it with certain restrictions).
ACT-R (MA&D/CMU)	Same as above.
Air MIDAS (SJSU)	Scheduling is deterministic and encapsulates both basic and domain-specific human behavior. Uses visual, auditory, cognitive, and motor channel resources on a numeric scale developed by McCracken and Aldrich (1984). Multi-tasking is possible based on resources available vs. resources required (e.g., structure allows two visual tasks to be performed concurrently provided sufficient resources are available.)
D-OMAR (BBNT)	Similar to Air MIDAS in some ways. The primary differences between D-OMAR and Air MIDAS are that the types of resources do not map directly and are quantified differently. The resources in D-OMAR represent five boolean values – the eyes (visual attention), ears (listening), mouth (speaking), a dominant and non-dominant hand (motor movements). As such, two visual tasks cannot be performed concurrently as they can with Air MIDAS.
A-SA (UIUC)	There is no explicit scheduler or task shedding mechanism in A-SA although the event timeline, an input to A-SA, can implicitly represent them

ACT-R Rice/UIUC and ACT-R MA&D/CMU

The scheduling of memory retrieval, visual attention, manual actions, and other skill-based tasks is performed by the central production system in ACT-R. The production system is intended to represent procedural memory, which stores associations between stimuli and response (in ACT-R parlance, condition and action) that indicate how to do something without consciously thinking about it. The production system detects patterns in the ACT-R buffers to determine what to do next. The production system is also capable of learning, but that aspect is discussed in section 5.9.

The production system is a rule set that operates by default in 50 ms time steps, though the timing of individual productions can be extended by retrievals or actions that take longer. A production rule (or simply "production") is an "if *condition* then *action*" pair. The *condition* specifies a pattern of chunks that must be present in the buffers for the production rule to apply. The *action* specifies some action to take. Multiple production rules can satisfy their respective conditions, but only one of those, the one with the highest utility, can execute its corresponding action. The production utility U_i is expressed as:

$$U_i = P_i G - C_i + \varepsilon$$ Production Utility Equation

where P_i is the probability of success that the production rule, if executed (or "fired" in ACT-R parlance), can achieve the current goal; G is the value of the goal (G is independent of the production rule, default value is 20); C_i is the cost (typically an estimate of the time from when the production is selected until the objective is finally completed); and ε is a noise parameter (e.g., if several production rules match the chunks in the buffers, there is a stochastic influence as to which production rule would be fired). The values of C_i and P_i can be set for each production rule or, as will be described section 5.9, can be learned from experience.

In practical terms, the production rules encapsulate both basic and domain-specific human behavior and thus represent task scheduling and priorities of the pilot. Thus, it is important that top-down, domain-specific behavior combined with bottom-up behavior (e.g., effort necessary to move visual attention elsewhere) be well understood by the modeler to reflect realistic task scheduling and task shedding by pilots. In addition, skill and experience of the modeler comes into play since the production parameters discussed above can be tuned to achieve stochastic or deterministic behavior (e.g., setting the production rules such that a pilot always gives immediate attention to ATC communication). As should be evident by this section and the section on memory (section 5.2), there is an art to ACT-R modeling (as there is with all HPM). It would be unlikely that two modeling teams would represent the production rules for a complex model such as the instrument approach effort in the same manner. In fact, this was exactly the case for the Rice/UIUC and MA&D/CMU teams as they each chose a unique production rule design for modeling visual attention (see section 5.5).

Historically, multi-tasking has not been an emphasis of ACT-R and most models created with ACT-R (including the instrument approach models) do not attempt to represent multi-tasking. However, with the release of version 5.0, which incorporates certain aspects of EPIC's perceptual and motor modules, ACT-R is capable of modeling parallel activity *with certain restrictions imposed by limitations observed in human performance*. First, modules in ACT-R such as the visual and manual modules operate independently of each other such that, for example, visual and manual tasks can be done in parallel as long as their initiation is staggered. This is due to a bottleneck in the central production system that results in, for example, a production rule for a visual task being delayed while a production rule(s) for a manual task is being fired (or vice versa). Since there is evidence for direct module-to-module interaction, ACT-R researchers acknowledge that ACT-R may require a modification to the framework to allow for this direct interaction. Second, each module can only do one job at a time due to the constraint on each module's buffer to only hold one chunk at a time (so two visual tasks cannot be done concurrently regardless of their resource demands).

The above restrictions, which are intended to simulate human limitations (Pashler, 2000), nonetheless result in a cumbersome process for multi-tasking between higher-level tasks (e.g., making a mode control panel entry while listening to an ATC communication). Multi-tasking at this higher level is better described as rapid interleaving between tasks as it requires ACT-R to continually switch between goals (for instance, in the above example, ATC communication must be broken into chunks to store in memory and each of these reflects a new goal) with each switch requiring the previous goal to be stored and then later retrieved from memory. Fortunately, there have been some techniques developed to facilitate this process, but it is still rather difficult compared to Air MIDAS and D-OMAR.

Air MIDAS

In Air MIDAS, pilot activities/tasks can be triggered for scheduling under two conditions. The first condition occurs when a value or event is encountered (typically through a monitoring scan

or as an auditory event) in the environment that has significance to the pilot. The second condition occurs based on the modeler-specified task decomposition (e.g., task 2 necessarily follows task 1). Once triggered, activities/tasks are then scheduled based on priorities, resources required, and resources available. The priorities and resources required for each activity are assigned by the modeler (often times with subject matter expertise input) whereas the resources available are calculated during run-time.

The scheduling mechanism in Air MIDAS considers both an estimate of the time it takes to complete a task and the availability of visual, auditory, cognitive, and motor channel (VACM) resources on a scale developed by McCracken and Aldrich (1984) and shown in Table 5. This theory assumes that the VACM channels are independent of each other. The scale for each channel ranges from 0 (no workload) to 7 (very high workload). The visual and auditory channels refer to the signal detection required for a task. The cognitive channel consists of the information processing and the motor channel is the physical action required to accomplish a task. One advantage to the VACM method is that it quantifies whether a task can be scheduled in a straightforward fashion by assuming the workload levels are additive within channels and independent between channels. For example, assume M=2 and V=4 for task A and V=2 and C=2 for task B. If task A is being performed, task B can only be scheduled to be performed concurrently with task A if each channel's total workload is less than 7, which it is for this example since M=2, V = 4+2 = 6, and C=2.

If the time or resources required to perform a task are not currently available, task priorities determine the outcome. Scheduling is accomplished through a set of queues of current, interrupted, and postponed activities to keep track of the activity order. Postponed and interrupted activities can be forgotten if there are too many procedures of the same type competing in the scheduler. The modeler can specify the number of active goals that can be kept in working memory at any given time. The active goals have procedures associated with their performance. By limiting the active goals the modeler can influence the number of procedures that can be kept current or pending based on first-in, first-out queuing process. If a higher priority goal is activated, which causes the available goal queue size to be exceeded, then the first goal to have entered the queue (i.e., the least recent) is aborted. In fact, this is how one of the error mechanisms in the taxiway operations effort was represented (see section 5.8). Note that for output purposes, the "forgotten" tasks are recorded even though they are never acted upon.

Multi-tasking is made possible through the same scheduling process with the requirement that the resource demands to perform parallel tasks do not exceed the human or system resources available. Numerical VACM scores above the threshold of 7 are typically not seen in Air MIDAS because the scheduler accounts for the excessive demand and postpones or sheds tasks accordingly. Practically-speaking, the visual demand of the instrument approach effort inhibits multi-tasking since a high value of visual resource is required for most tasks. For example, the Air MIDAS team assigned a value of 5.9 to the visual channel during the task of monitoring [i.e., "Visually Read (symbol)" from Table 5]. This means the remaining visual resource available is only 1.1 (i.e., 7 – 5.9 = 1.1) for other tasks to be performed in parallel. As can be seen in Table 5, most tasks have visual resource requirement higher than 1.1. Thus, the monitoring scan must be suspended/interrupted in order to initiate most other tasks, thereby inhibiting multi-tasking.

Lastly, it should be noted that Air MIDAS differs in three ways from ACT-R in regards to multi-tasking. First, there are no architectural bottlenecks in Air MIDAS analogous to the central production system bottleneck of ACT-R (see ACT-R section for explanation). Second, multiple tasks using the same channel resource are permitted provided the sum of their demands do not exceed the channel threshold (e.g., two low-demand visual tasks can occur concurrently). Third, the Air MIDAS scheduler is deterministic rather than stochastic.

Table 5. VACM Values and Descriptors (McCracken and Aldrich, 1984)

Value	Visual Scale Descriptor	Value	Auditory Scale Descriptor
0.0	No visual activity	0.0	No auditory activity
1.0	Register/detect image	1.0	Detect/register sound
3.7	Discriminate or detect visual differences	2.0	Orient to sound, general
4.0	Inspect/check (discrete inspection)	4.2	Orient to sound, selective
5.0	Visually locate/align (selective orientation)	4.3	Verify auditory feedback (detect occurrence of anticipated sound)
5.4	Visually track/follow (maintain orientation)	4.9	Interpret semantic content (speech)
5.9	Visually Read (symbol)	6.6	Discriminate sound characteristics
7.0	Visually scan/search/monitor (continuous/serial inspection, multiple conditions)	7.0	Interpret sound patterns (pulse rates, etc.)
Value	**Cognitive Scale Descriptor**	**Value**	**Motor Scale Descriptor**
0.0	No cognitive activity	0.0	No motor activity
1.0	Automatic (simple association)	1.0	Speech
1.2	Alternative selection	2.2	Discrete actuation (button, toggle, trigger)
3.7	Sign/signal recognition	2.6	Continuous adjusting (flight control, sensor control)
4.6	Evaluation/judgment (consider single aspect)	4.6	Manipulative
5.3	Encoding/decoding, recall	5.8	Discrete adjusting (rotary, vertical thumbwheel, lever position)
6.8	Evaluation/judgment (consider several aspects)	6.5	Symbolic production (writing)
7.0	Estimation, calculation, Conversion	7.0	Serial discrete manipulation (keyboard entries)

D-OMAR

In D-OMAR, scheduling and multi-tasking share some similarities to Air MIDAS. For example, pilot activities/tasks are triggered for scheduling under essentially the same two conditions from above: 1) In response to values or events in the environment that have significance to the pilot, or 2) Based on the sequence of tasks resulting from normal pilot procedures and basic human behavior. If the resources required for the task are available, the triggered activity/task can begin immediately, possibly in concurrence with the current task. If the resources required are not available, task priorities assigned by the modeler determine if the current task should be interrupted to start the new task or if the new task should be postponed. In some instances, task priorities are based on well-accepted policy (e.g., intra-crew conversation should be interrupted when an ATC communication is heard over the radio). The primary differences between D-OMAR and Air MIDAS are that the types of resources do not map directly and are quantified differently. In Air MIDAS, the visual, auditory, cognitive, and motor resources are scaled numerically from 0 to 7 for each task element so that two low-demand visual tasks, for example, can occur concurrently. In contrast, the resources in D-OMAR represent five boolean values – the eyes (visual attention), ears (listening), mouth (speaking), a dominant and non-dominant hand (motor movements). In other words, each resource can only be allocated to one task at a time (e.g., an operator can only listen to one conversation at a time). In effect, the "eyes" resource

inhibits multi-tasking in most situations since visual attention is necessary for most tasks. In addition, it is assumed that the cognitive channel is implicitly coupled with the other resources and is therefore not resource-limited.

A-SA

There is no explicit scheduler or task shedding mechanism in A-SA although the event timeline, an input to A-SA (rather than an output as it is for the other modeling tools), can implicitly represent them.

5.4 Crew interactions

This section discusses the method for modeling crew interactions for each of the modeling teams. The level of fidelity at which the crew interactions should be represented is similar to the external environment issues discussed earlier. In fact, from a single-operator perspective, the crew interactions are essentially a subset of the external environment. By identifying what question the modeling effort is trying to answer, insight can be gained into level of fidelity requirements. If, for example, new procedures are being developed specifically to address crew resource management issues, then the modeling effort should emphasize high fidelity crew interactions because poor performance of one crew member could require corrective actions by the other crew member – similar to the feedback mechanism between pilot and aircraft for the external environment mentioned earlier. On the other hand, if it is a new display such as SVS, which does not directly affect procedures for crew interactions, then lower fidelity modeling, such as scripting the interactions of one of the crew members, should be sufficient.

Table 6. Crew Interactions Synopsis

Modeling tool/team	Crew interaction representation
ACT-R (Rice/UIUC)	Only modeled the pilot-flying. Interactions with the pilot-not-flying (and ATC) were scripted.
ACT-R (MA&D/CMU)	Used ACT-R to model the pilot-flying and IMPRINT to model the pilot-not-flying (lower fidelity).
Air MIDAS (SJSU)	Both the pilot-flying and pilot-not-flying were simulated at equivalent levels of fidelity by Air MIDAS.
D-OMAR (BBNT)	Both the pilot-flying and pilot-not-flying were simulated at equivalent levels of fidelity by D-OMAR.
A-SA (UIUC)	Only modeled the pilot-flying. Interactions with the pilot-not-flying (and ATC) were scripted.

ACT-R Rice/UIUC

The Rice/UIUC team only modeled the pilot-flying. Interactions with the pilot-not-flying (and ATC) were scripted.

ACT-R MA&D/CMU

The MA&D/CMU team represented crew interactions using IMPRINT. As mentioned earlier, IMPRINT is itself a human performance modeling tool. Thus, the MA&D/CMU team chose to use ACT-R to model the pilot-flying and IMPRINT to model the pilot-not-flying. This allowed interactions between the crew to occur in a non-scripted manner, although the pilot-not-flying's behavior modeling was lower fidelity by comparison. In any case, this proved to be a good

compromise between the required fidelity of the pilot-flying and perhaps unnecessary complexity for the pilot-not-flying.

Air MIDAS

Recent developments in Air MIDAS have emphasized multiple-operator environments so that modeling studies can investigate the closed-loop behavior (e.g., between pilots of the same aircraft as well as pilots of different aircraft) that would be essential to the understanding of new operational concepts such as the self-separation of aircraft. These developments have been leveraged for the instrument approach modeling effort to simulate both the pilot-flying and pilot-not-flying at equivalent levels of fidelity. This effort required detailed model input for both pilot positions in terms of external environment, procedures, task priorities, etc. The resulting interactions between the pilots thus provided realistic closed-loop behavior, with latency being the most common effect, while adhering to attentional and workload constraints.

D-OMAR

The modeling framework of D-OMAR lends itself quite well to modeling multiple human operators. For both NASA modeling efforts, the D-OMAR team modeled the pilot-flying and pilot-not-flying at equivalent levels of fidelity. Similar to Air MIDAS, this effort required detailed model input for both pilot positions in terms of external environment, procedures, task priorities, etc. The resulting interactions between the pilots thus provided realistic closed-loop behavior.

A-SA

Crew interactions in A-SA are scripted and typically represented as an auditory event in an event timeline. The salience of an auditory event (which is typically assigned maximum weights in A-SA) captures the pilot's attention. Whether pilot SA is improved or degraded depends on the relevance of the crew interaction to tasks at hand. Interestingly, an irrelevant interaction would degrade SA faster than if there had been no interaction at all.

5.5 Visual attention

Developing a capability to predict visual attention was a key aspect of the instrument approach modeling effort for most of the teams. Visual attention describes how the pilot moves his attention between displays and between items on a display to extract information from the external environment. During the monitoring task, the pilot periodically views each display in a manner that we refer to in this paper (using D-OMAR parlance) as a *background scan pattern*. Interestingly, each modeling team took a unique approach to modeling the background scan pattern. During the background scan, certain values encountered from the external environment trigger reactive/procedural tasks (as do certain auditory events), which take precedence over the background scan pattern and direct visual attention elsewhere. Once the reactive/procedural task(s) are finished, the background scan is resumed. The resultant visual attention allocation is thus the combined effects of the background scan pattern and the reactive/procedural scans, typically discussed in terms of fixations and dwells. A "dwell" is defined as the time period during which a fixation or series of continuous fixations remain on a particular display.

Note that none of the teams attempted to emulate "perception" or "pattern recognition". Text and numeric values were simply assumed to be read into memory when a fixation of sufficient duration occurred. For non-textual and non-numeric information, each of the modeling teams devised ways to convert the context/content of the information into text or numeric parameters within the model. For example, the out-the-window representation of "runway-in-sight" was typically converted to a boolean variable. Furthermore, none of the teams attempted to address

visual workload issues related to data fusion. For example, the SVS display presents runway alignment rather easily via a 3D rendering of the runway and flight path predictor whereas the traditional displays requires piecing together several display items to develop an equivalent mental picture.

Table 7. Visual Attention Synopsis

Modeling tool/team	Visual attention representation
ACT-R (Rice/UIUC)	Background scan based on the information needs of the pilot as determined by the team's internally-generated task analysis and the effort of acquiring information from the displays.
ACT-R (MA&D/CMU)	Background scan employed the learning mechanism in ACT-R to learn both the information needs of the pilot and the effort of acquiring information from the displays.
Air MIDAS (SJSU)	Background scan derived from eye-tracking data and augmented with scan policy for certain segments of flight.
D-OMAR (BBNT)	Background scan consists of three scan types – basic PFD scan plus heading, a ND scan, and out-the-window scan. Frequency of these scan types were determined by calibrating with eye-tracking data.
A-SA (UIUC)	Visual attention in A-SA is represented by the SEEV model. SEEV asserts a pilot's attention is *influenced* by two factors: 1) bottom-up capture of salient or conspicuous events, and 2) top-down selection of valuable event information in expected locations at expected times. In contrast, attention is *inhibited* by the bottom-up level of effort required, which encompasses both the effort of moving attention elsewhere and the effort required due to concurrent cognitive activity.

ACT-R Rice/UIUC

For the instrument approach effort, the primary goal of the Rice/UIUC team was to model visual attention as thoroughly as practical with minimal tuning of the default ACT-R parameters. In order to do this, they needed to model bottom-up and top-down factors. The primary bottom-up factor was the layout of the displays and the associated effort of acquiring information from the displays. This was particularly pertinent when predicting the background scan pattern when SVS was available since the SVS display contained redundant information from the PFD (e.g., altitude, air speed) and the ND (e.g., heading). With SVS, pilots can expend less effort (minimize cost in ACT-R parlance) if they attend to the nearest redundant information. Although ACT-R has a mechanism for updating the cost/effort during run-time (used by the ACT-R MA&D/CMU team below), the Rice/UIUC team estimated the cost/effort a priori based on the physical layout of the displays and saccade (rapid voluntary eye movements used to move from one fixation to another) latency and accuracy.

The key top-down factor in the model was the information needs of the pilot based on the ACT-R team's internally-generated task analysis. This task analysis was decomposed into *aviate*, *navigate*, and *manage systems* high-level tasks and corresponding monitoring loops:

- The *aviate* monitoring loop consisted of monitoring the speed, altitude, flap settings, and vertical speed.

- The *navigate* monitoring loop consisted of monitoring the flight trajectory, heading, location along flight path, distance to next waypoint.
- The *manage systems* monitoring loop consisted of monitoring the MCP to confirm entries and monitoring the modes of the flight management system (FMS) to confirm LNAV and VNAV.

How frequently the three monitoring loops are executed relative to each other identifies the information needs of the pilot for the monitoring task. However, estimating the frequencies of the respective monitoring loops using subject matter experts is unlikely to be accurate. Instead, the Rice/UIUC team calibrated the frequencies heuristically until they achieved a good match with visual attention allocation data from one segment of one SVS run from the Santa Barbara part-task simulation. The complete process is as follows:

> Calibrating the frequency of "aviate", "navigate", "manage systems" monitoring loops (with respect to each other) causes an adjustment to the information needs of the pilot, which causes an adjustment to the background scan pattern, which, combined with event-driven, reactive/procedural scans, results in the overall allocation of visual attention. The calibration process is repeated until a good fit of visual attention is achieved.

Note that this was the only calibration performed. The remaining ACT-R runs were generated without parameter adjustment to predict visual allocation attention.

ACT-R MA&D/CMU

Compared to the ACT-R Rice/UIUC team, the ACT-R MA&D/CMU team took a fundamentally different approach to modeling visual attention by enabling learning in ACT-R. The learning mechanism had two components to it. The first component learned the "information needs" of the pilot during the background scan by assigning "success" to rewarding information. The modelers defined rewarding information on two levels – first, that the information was changing, and second, that the information resulted in the pilot taking some sort of action (e.g., reactive/procedural tasks). Information on the display that is neither changing nor results in action is a "failure". (By comparison, the Rice/UIUC team used calibration to identify the information needs.) In addition, if the needs of the pilot changed over time (e.g., over a flight segment), there is a stochastic factor that occasionally enables previously "failed" pieces of information to be chosen – giving them another chance to prove their usefulness. This stochastic effect coupled with the fact that "success" decays according to a power law process means that if the information needs continuously change then the background scan pattern would continuously adapt.

The second component, which only applied when SVS was available, was specifically designed to address why some of the subject pilots from the Santa Barbara part-task simulations relied mostly on the ND and PFD whereas other pilots relied on SVS (which in addition to the 3D rendering of the terrain also contains an overlay of redundant information such as air speed and altitude that are also on the PFD). The MA&D/CMU team hypothesized that once visual attention was directed toward a particular display, the availability of redundant information (due to the presence of SVS) would increase the probability that the next display item needed would also be found on that display. Thus, a pilot could make his/her background scan pattern more efficient by choosing the next display item (assuming it is one of the redundantly available pieces) from the currently attended display since less time/effort is required to move a small distance within the current display than to move to a different display altogether.

To implement this, production rules were developed to levy a "cost" corresponding to the time/effort of moving attention to the next display item – the further the distance, the higher the cost. Two production rules were necessary to apply the cost mechanism. One production rule

specified the traditional displays (ND and PFD) as the location of redundantly available information whereas the other specified SVS. Initially, the production rule selected was determined purely randomly (with each production rule have a 50% chance of being selected) corresponding to which display type, either traditional or SVS, would win the pilot's attention. However, over time, as one display randomly received more attention (analogous to a coin toss resulting in multiple "heads" in a row), the time/effort/cost of the scan would incrementally decrease since the next display item needed would more likely than not be found on that same display. The reduced cost would then incrementally increase the probability of that particular display type being selected in the future, resulting in convergence to an all-or-nothing use of the display type – either it would be all traditional and no SVS or vice versa. Of course, this all-or-nothing convergence refers to the background scan patterns. The total visual attention allocation is the combined effects of the background scan pattern and the reactive/procedural scans.

A key benefit to this learning methodology is that it is possible to model and evaluate behavior in an environment for which no human performance data exist. The allocation of attention and consequent use of the interface is a *predictive* model of human behavior, rather than a descriptive model that simply emulates existing data.

As mentioned earlier, the IMPRINT representation of the ND in the external environment did not capture the directional aspects of the flight. In the real world, a pilot would scan the aircraft's progress with respect to waypoints and attend to the ND more frequently prior to and during a heading change to ensure that the aircraft is maintaining its flight path. Without this effect modeled, the simulated pilot would attend to the ND less frequently than in the real world.

Air MIDAS

The Air MIDAS team took a completely different approach to visual attention compared to the two ACT-R teams. The modeling of visual attention started with four representative scan patterns of the pilot task based on eye-tracking data collected from the Santa Barbara part-task simulations. This data was represented in terms of the percentage of time the subject pilots fixated on the various flight displays. The displays where the subjects spent significant percentages of time were the PFD, ND, out-the-window, and SVS (when available to the subjects). The four patterns represented VMC without SVS, VMC with SVS, IMC without SVS, and IMC with SVS. In addition, the part-task scan patterns were slightly modified to account for small differences in display mapping between the part-task layout and the layout of the 757 simulated by Air MIDAS. Next, a scan "policy", based on an aircraft operating manual for the pilot-flying and pilot-not-flying, specified how the scan pattern should be modified to include the out-the-window view in the scan depending on whether the runway was in sight and whether the aircraft had approached decision altitude. The scan policy assumptions were the last step necessary to construct the background scan pattern prior to run-time.

As compared to the ACT-R Rice/UIUC team's approach, an interesting aspect of the Air MIDAS approach is that the behavior of the background scan pattern closely replicates the required input frequency of the "aviate/navigate/manage systems" monitoring loop components of the ACT-R Rice/UIUC model (recall that this was calibration mechanism used by the Rice/UIUC team). Ignoring SVS for the sake of discussion, the aviate monitoring loop in the ACT-R Rice/UIUC model primarily scans the PFD; the navigate monitoring loop scans the ND; and the manage systems monitoring loop primarily scans the MCP. By comparison, the background scan pattern of Air MIDAS specifies directly what percentage of time the PFD, ND, and MCP should be scanned. Thus, the background scan pattern in Air MIDAS acts as a surrogate for the monitoring loop task decomposition structure by implicitly representing the "aviate/navigate/manage systems" frequency of execution. An advantage to the Air MIDAS approach is that if these monitoring loops required subtle changes to their frequency of execution as a function of flight

segment, then the effect is captured automatically. On the other hand, the Air MIDAS approach is more dependent on eye-tracking data to define the background scan pattern (the ACT-R Rice/UIUC team needed only one flight segment from one scenario) and might not easily generalize to a different cockpit setup or scenario.

Another aspect of visual attention that the Air MIDAS team modeled was the failure to acquire information during fixation if the duration of the fixation was below a threshold. The threshold was determined heuristically to provide a 10% failure in fixation and applied to background scans, but not reactive/procedural scans. Thus, 10% of the fixations from the background scan acquired no information from the external environment while directed fixations always succeeded.

Lastly, the Air MIDAS team assumed that the speed and altitude values from the SVS were not used to trigger any reactive/procedural task. This is due to the original concept of operations (ConOps) for SVS that stated that SVS was not intended for flight directive or commanding purposes. The Air MIDAS team chose to embed this ConOps into their model. In contrast, the other modeling teams explored the utility of violating the ConOps by using the SVS for flight directive or commanding purposes to demonstrate the efficiency of attending to redundant information overlaid on the SVS display.

D-OMAR

The D-OMAR team took yet another approach to modeling the background scan pattern compared to the other teams. The D-OMAR team developed a "basic" scan that read speed, altitude, altitude rate, and heading. In the baseline condition (no SVS), the first three display items were read from the PFD and the latter item was read from the ND. (It is worthwhile to note that although the location of the heading display item was on a separate display, the effort to acquire it was minimal when scanning the PFD because it was situated directly below the PFD at the top of the ND.) In addition to the basic scan, a "horizontal situation" scan of the ND was developed to look at the current flight segment, next flight segment, heading, display range setting, and distance to the active waypoint from the ND. Lastly, the out-the-window view was scanned periodically, but at a much lower frequency.

With SVS, the basic scan was assumed to be read from either the PFD/ND or SVS in an alternating fashion. In addition, the view of the runway was added to the scan of SVS prior to descending below the cloud cover. The horizontal situation scan and out-the-window scan were unchanged from the baseline.

The intervals between scans for each of the scan types (basic, horizontal situation, and out-the-window) were controlled by three unique variables. These variables were tuned against the Santa Barbara part-task simulation data such that the three scan types collectively utilized all the pilot's available time (assuming a segment of the flight when no other tasks were being performed). This defined the background scan pattern. As with the other models, scanned display items were used to trigger reactive/procedural tasks, which took precedence over the background scan pattern and directed visual attention elsewhere. Lastly, transitions from display to display were not specified by the modeler, but rather determined based on trying to maintain the desired intervals while interleaving between the background scan and reactive/procedural scans.

A-SA

Visual attention in A-SA is represented by the SEEV model. As discussed in section 4.5, SEEV asserts a pilot's attention is *influenced* by two factors: 1) bottom-up capture of salient or conspicuous events (visual or auditory), and 2) top-down selection of valuable event information in expected locations at expected times. In contrast, attention is *inhibited* by the bottom-up level

of effort required, which encompasses both the effort of moving attention elsewhere and the effort required due to concurrent cognitive activity. Under normal conditions, a skilled pilot should not be influenced by salience or inhibited by the level of effort required.

The probability of attending to a particular flight deck display or out-the-window view, *P(A),* is represented by the following equation:

$$P(A) = sS - efEF + (exEX * vV)$$

where coefficients in upper case (S for **salience**, EF for **effort**, EX for **expectancy**, and V for **value**) describe the properties of a display or environment while those in lower case describe the weight assigned to those properties in directing of an operator's attention. The values for these coefficients were estimated by the A-SA team.

For the instrument approach modeling effort, visual attention allocation was predicted for four flight segments of the approach. SEEV parameters were estimated for each of the four segments. They modeled the **effort** of moving attention from one display to another by assuming that such effort is monotonically related to the distance between displays. The **value** of a display was equal to the value of the task served by the display multiplied by the relevance of that display to the task in question. The value of the task was based on the priority of the *aviate* and *navigate* high-level tasks to the flight segment. For example, when the pilot took manual control of the aircraft in the last segment, these priorities increased dramatically. The **expectancy** for information contained on the display was determined by bandwidth (rate of change) of the display. Note that the A-SA team did not model the **salience** of events because there were few salient events to capture during approach.

5.6 Workload

This section discusses the capabilities of the modeling tools for quantifying workload.

Table 8. Workload Synopsis

Modeling tool/team	Workload representation
ACT-R (Rice/UIUC)	No explicit measures of workload available as an output, but. workload constraints are still present in the system (e.g., if task demands exceed available resources, some tasks or subtasks will not be completed).
ACT-R (MA&D/CMU)	See above.
Air MIDAS (SJSU)	Workload estimates based on the VACM resource theory of workload. Average workload over a flight segment was calculated to compare baseline condition to SVS.
D-OMAR (BBNT)	No explicit measures of workload available as an output, but. workload constraints are still present in the system.
A-SA (UIUC)	No attempt to model the workload of action-related phenomena. Does model the effect of cognitive load on attention and SA.

ACT-R Rice/UIUC and ACT-R MA&D/CMU

For the instrument approach modeling effort, there were no explicit measures of workload available in the models developed by Rice/UIUC or MA&D/CMU other than the visual attention allocation already discussed. The lack of explicit workload metrics is more of a bookkeeping issue rather than a theoretical one. In the course of a run, ACT-R essentially characterizes the human operator's state every 50 ms including the resources used so it would be possible to add the required code to generate some type of workload metric based on the percentage of time human resources are allocated to tasks and activities and the percentage of time these resources are unused and available. Workload constraints are still, however, present in the system. If task demands exceed available resources, some tasks or subtasks will not be completed. Thus, excessive workload will produce performance errors in ACT-R models, but explicit prediction of how close to this point a model is operating is not straightforward.

Air MIDAS

Workload in Air MIDAS is based on the scale in Table 5 to provide workload estimates compatible with the VACM resource theory of workload. Numerical VACM scores above the threshold of 7 per channel are typically not seen in Air MIDAS because the scheduler accounts for the excessive demand and postpones or sheds tasks accordingly.

Average workload is a useful parameter for characterizing relative differences between a baseline condition and a condition that represents new procedures or new displays such as SVS. Average workload over a time period of interest (such as a flight segment) is expressed by:

$$WL_{i\ avg} = \sum_{j=1}^{n} \frac{WL_{i,j}\Delta t_{i,j}}{t_{total}}$$

where i is the VACM channel, j represents each task element executed by the model (e.g., move hand to mode control panel, read value from display), n is the number of task elements executed over the total time duration, $WL_{i,j}$ is the VACM workload of each task element, $\_t_{i,j}$ is the duration of the task element, and t_{total}.is total time duration to average across. Note that the $WL_{i,j}$ values are VACM channel inputs assigned by the modeler for every possible task element to be performed whereas $\_t_{i,j}$ is calculated during run-time using equations such as Fitt's law.

A similar equation was used to categorize workload by type of procedure/decision such as the "land or go-around" decision, performing the landing checklist, and performing the background scan pattern.

D-OMAR

For the instrument approach modeling effort, there were no explicit measures of workload in D-OMAR other than the visual attention allocation already discussed. As with ACT-R, this is more of a bookkeeping issue rather than a theoretical one. D-OMAR could be coded to output the percentage of time each of the 5 resources (eyes, listening, speaking, dominant hand, and non-dominant hand) is utilized as a workload metric for comparison between the baseline and SVS condition.

A-SA

A-SA makes no attempt to model the workload of action-related phenomena. However, it does attempt to model the effect of cognitive load on attention and SA. For example, for the taxiway operations model, cognitive load for an event is incurred for some time after the event is first processed. As a new event occurs, the attentional resources available to process it are inversely

proportional to the sum of residual attentional weights from previous events, which effectively enables cognitive load to modulate current attentional control.

5.7 Situation awareness

Situation awareness is often discussed in terms of Endsley's three levels (1988):

- Level 1 SA – Perception of the elements in the environment
- Level 2 SA – Comprehension of the current situation
- Level 3 SA – Projection of future status

For the instrument approach modeling effort, the background scan patterns developed by the modeling teams address Level 1 SA. One can assume that if the pilot interleaves his background scan with other tasks requiring attention, Level 1 SA would be maintained. With the exception of A-SA, the modeling teams did not attempt to predict Level 2 & 3 SA as an explicit metric in their models. However, the ACT-R Rice/UIUC team did attempt to model the go-around decision based on an integration of information that included trajectory alignment components as well as human predispositions. This decision could be treated as an implicit form of SA. (Note that the implementation of the go-around decision was incomplete at the time they delivered the instrument approach modeling final report.)

For the taxiway operations modeling effort, the Air MIDAS and D-OMAR modeled both pilot-flying and pilot-not-flying. Again, although no explicit measures of SA were generated, the activity of the pilot-not-flying had an impact on the crew interactions, which had an impact on the SA of the pilot-flying, which had an impact on whether the pilot made the correct turn at each intersection encountered. This again could be treated as an implicit form of SA. See section 5.8 for more discussion on this topic.

A-SA

A-SA was specifically designed to address SA. The SA component of A-SA is discussed in the belief updating module in section 4.5. The model structure is designed such that the value of SA is used to guide subsequent attentional scanning as indicated by the following equation from the taxiway operations version of the model:

$$W = C + SA * V$$

where W is the attentional weight of a stimulus/event, C is the conspicuity of the event, SA is the current value of SA (between 0 and 1), and V is the information value. Thus, good SA guides attention towards relevant information whereas poor SA guides attention towards conspicuous, but less relevant information.

A-SA also utilizes SA for error prediction – see section 5.8.

5.8 Error prediction

Modeling capabilities for error prediction come in many forms. In Leiden et al (2001), error taxonomies and the error prediction capabilities of a range of human performance modeling tools are presented. In general, error prediction can be characterized by the capabilities of the modeling tool. For example, memory-related errors require a modeling tool that represents memory interference and decay. Errors due to flight crew interactions require high-fidelity modeling of the pilot-flying and pilot-not-flying (scripted behavior would be ineffectual). Moreover, error prediction in human performance modeling almost always results in some type of priming to allow the error to occur. The reason for this is that most complex environments such as aviation have procedures in place to deal with commonplace human errors. For example, the readback of a clearance ensures both that the pilot understood the controller correctly and that the controller (while listening to the readback) issued the intended clearance. It is still possible that the pilot would perform the clearance incorrectly, but there needs to be a more complex layer in the error chain to manifest itself. Thus, in order to prime an error in the model, the modeler must have an understanding of realistic events and activities (and their frequency of occurrence) so that the types of errors produced represent realistic errors beyond the commonplace errors mitigated by checks in routine procedures. Otherwise these errors may never occur at all in simulation or occur so frequently as to lack the face validity necessary to be considered meaningful. This aspect of the modeling is most likely the crux in terms of representing accurate error rates since the information is often difficult to quantify and is therefore often crudely estimated.

Table 9. Error Prediction Synopsis

Modeling tool/team	Error prediction representation
ACT-R (Rice/UIUC)	Developed five primary decision making strategies that a pilot employs when approaching a taxiway intersection to decide which taxiway to take. Hypothesized that the different strategies (each with different probability of error) would have a probability of selection based on the decision time horizon.
ACT-R (MA&D/CMU)	Represented errors due time-based decay or activation noise that resulted in a failure to recall a chunk holding turn information. The resulting error would be a missed turn. Also, represented errors that resulted in the wrong chunk being recalled. This would cause the model to schedule a turn on the wrong taxiway or in the wrong direction on the correct taxiway.
Air MIDAS (SJSU)	Represented three types of error mechanisms. 1) Pilots forgetting to complete a procedure as a result of multi-tasking between too many procedures at a given time. 2) Loss of information in working memory. 3) Worldview inconsistency between two operators. For example, this error could occur when the ATC erroneously assumed the pilots had accurately received taxi clearance information.
D-OMAR (BBNT)	Represented errors based on: 1) Episodic memory – an element of autobiographical memory and a source for habit-based actions. 2) Context-based expectation based on partial knowledge – the pilot-flying knows the location of the destination concourse gate and based on this knowledge may reasonably have an expectation that the next turn will take them on the shortest route to that gate. 3) Remembrance – the pilot-flying remembers a correct or incorrect taxi sequence issued by the ground controller.
A-SA (UIUC)	Error prediction is provided in terms of errors of SA. The value of SA (between 0 and 1) was assumed to reflect the probability that the pilot when reaching a decision point at a taxiway intersection would select the correct response. If the pilot failed to recognize the correct response, then the default responses (modeler-specified) such as "when in doubt turn toward the terminal" or "when in doubt go straight" would represent the default decision.

ACT-R Rice/UIUC

For the taxiway operations modeling effort, the Rice/UIUC team [using subject matter expert (SME) input] developed five primary decision making strategies that a pilot employs when approaching a taxiway intersection to decide which taxiway to take. They hypothesized that the different strategies would have a probability of selection based on the decision time horizon – the

time available until a turn must be executed at an intersection. For example, making the decision to turn based on guessing would be fast, but inaccurate. Thus, the pilot is likely to only choose this strategy when there is insufficient time available for the other strategies. The strategies are summarized below in order of the decision time horizon required:

1. Guess randomly – a very fast, but inaccurate strategy.
2. Remember the correct clearance – while fast, this strategy is increasingly inaccurate as memory decays.
3. Make turns toward the gate – somewhat slower than the first strategy, but 80.7% accurate (based on their internally-generated analysis for 9 airports in the USA).
4. Turn in the direction that reduces the larger of the X or Y (cockpit-oriented) distance between the aircraft and the gate. 93%[§§] accurate based on their analysis.
5. Derive from map/spatial knowledge – this is the slowest strategy available.

Rather than running the ACT-R model to predict specific instances of error, the Rice/UIUC team took a more global approach to the problem. Production rules were generated for each of these strategies where each strategy had a minimum time horizon (based on the order above) that was required for it to be selected. Furthermore, the probability of success component of the production utility (see section 5.3) was set to how accurate the strategy was deemed to be. For strategy #1, the accuracy was based solely on the taxiway geometry. For example, if the intersection had three choices weighted equally, the probability of success would be 0.33. For #3 and #4, the values corresponded to 0.807 and 0.93 as indicated above. For #2 and #5, the probability of success was based on the declarative memory component of ACT-R to recall these chunks.

The ACT-R model was run in Monte Carlo fashion to predict the probability of selection of these strategies for time horizons varying from 0.8 - 10.6 s in 0.2 s increments. Thus, the error rates were a function of time horizon such that:

$$Probability\ of\ error_k = 1 - \sum_k \sum_j St_{jk} P_j$$

where k is the time horizon in 0.2 s increments, j is the decision strategy, St_{jk} is the probability of strategy selection for each j and k, P_j is the probability of a strategy's success. For example, their results indicated that for a 4.2 s time horizon, the probabilities of selecting strategy #3 and #4 were each about 50%. Thus, the probability of error is about 13% [=1 – (0.93*0.5+ 0.807*0.5)].

ACT-R MA&D/CMU

For the taxiway operations modeling effort, the MA&D/CMU team predicted taxiway errors based on memory decay and interference. The taxiways turns were represented by two chunks each, one chunk indicating the name of the taxiway and the other chunk holding the direction to turn. At the appropriate time in the simulation, the ACT-R pilot was called for the initial memorization of the list of taxiways. Later in the sequence, the aircraft approached a taxiway intersection and the ACT-R pilot attempted to recall the list of taxiways. If the attempt was successful, the correct taxiway was followed. If unsuccessful, the following error types could be produced by the model:

- Errors due time-based decay or activation noise that resulted in a failure to recall a chunk holding turn information. The resulting error would be a missed turn.

[§§] This value was calculated by the first author based on the bar chart data from Figure 2 of the ACT-R Rice/UIUC chapter and thus may be prone to interpretation error.

- Errors due to interference, similarity-based partial matching, priming, or activation noise that resulted in the wrong chunk being recalled. This would cause the model to schedule a turn on the wrong taxiway or in the wrong direction on the correct taxiway.

Air MIDAS

For the taxiway operations modeling effort, the Air MIDAS team represented three types of error mechanisms. The first error type represented the pilots forgetting to complete a procedure as a result of multi-tasking between too many procedures at a given time. The occurrence of this error was modeled by causing events in the simulation to occur at approximately the same time such that postponed and interrupted activities would be forgotten because there were too many procedures competing in the scheduler (see section 5.2). The second error type represented errors due to loss of information in working memory. These two error types could be combined to replicate a wrong turn. For example, if the pilot-not-flying had his head down while approaching a required turn due task interruption and forgetting to resume the task in a timely manner, then the pilot-flying could make an erroneous turn based on his failure to recall the correct turn from working memory. The third type of error represented worldview inconsistency between two operators. For example, this error could occur when the ATC erroneously assumed the pilots had accurately received taxi clearance information.

Two manipulations were used to test the impact of memory demands on crew performance – the working memory decay rate and capacity. Decay rate was set for information loss at either 12 seconds or 24 seconds from the time of perception or recall and the memory capacity was either 5 chunks or 10 chunks.

D-OMAR

For the taxiway operations modeling effort, the D-OMAR team represented taxiway errors by priming the model such that reasonable scenarios were constructed in which the pilot-not-flying was occupied with event-driven tasks and unable to provide the prompt to the pilot-flying to turn at an intersection. Without the explicit prompt, the decision of the pilot-flying to turn at an intersection was based on other intentions:

1. *Episodic memory* – an element of autobiographical memory and a source for habit-based actions. Experience from similar situations provides a response that has worked well in the past.
2. *Context-based expectation based on partial knowledge* – the pilot-flying knows the location of the destination concourse gate and based on this knowledge may reasonably have an expectation that the next turn will take them on the shortest route to that gate.
3. *Remembrance* – the pilot-flying remembers a correct or incorrect taxi sequence issued by the ground controller.
4. *Explicit prompt* by the pilot-not-flying based on written taxi instructions. Nominally, the intention of the pilot-flying and the prompt from the pilot-not-flying are in agreement.

The priority assigned to these intentions and timing of events determined the intention selected by the model. The timing and priorities were varied by hand to explore their impact on producing errors in the taxi sequences.

A-SA

For the taxiway operations modeling effort, error prediction is provided by A-SA in terms of errors of SA. (SA itself is discussed in sections 4.5 and 5.7.) In other words, the value of SA (between 0 and 1) was assumed to reflect the probability that the pilot when reaching a decision

point at a taxiway intersection would recognize the correct response option and behave appropriately as a result. If the pilot failed to recognize the correct response option, then the default responses (modeler-specified) such as "when in doubt turn toward the terminal" or "when in doubt go straight" would represent the default decision. If more than two responses applied to a decision point, the response selected would be chosen at random. The equation for the probability of error can be expressed as:

$$Probability\ of\ error = 1 - [SA + (1 - SA)P_{avg}]$$

where P_{avg} is the average probability of all default responses being correct. Because these default responses tend to be reasonably accurate, even with low SA the probability of making an incorrect decision is lower than one would first suppose (e.g., probability of error is 20% for the low SA condition). It is important to note that the error prediction capability within A-SA (and all the other models for that matter) is quite domain-specific.

5.9 Learning

Learning or adapting is a key feature of certain HPM tools. Whether learning is important to the modeling efforts conducted under the NASA HPM Project appears to have two schools of thought. On one hand, part-task simulation data collected for the instrument approach effort (for the expressed purpose of HPM calibration and validation) was gathered from subject pilots with considerable experience. Given this experience, the notion of their behavior changing in the baseline condition (no SVS) over a few minutes of simulation in reaction to learning or adaptation is unlikely. On the other hand, the subject pilots had limited training with SVS (relative to the number of hours of flight experience) so one could hypothesize that their behavior with respect to SVS perhaps had not yet reach steady-state in terms of adaptation. Of the four HPM tools, Air MIDAS, D-OMAR, and A-SA do not address learning. In contrast, learning is often cited as one of ACT-R's strengths and the MA&D/CMU team chose to enable it in their modeling effort whereas the Rice/UIUC team chose to disable it.

ACT-R MA&D/CMU

The MA&D/CMU team enabled learning in ACT-R by utilizing two complementary components to determine the background scan pattern (see section 5.5 for further details). This can be summarized as follows: For the first component (referred to as the *information need* component), the information needs of the pilot were learned by assigning "success" to rewarding information on the displays. The modelers defined rewarding information on two levels – first, that the information was changing, and second, that the information resulted in the pilot taking some sort of action (e.g., reactive/procedural tasks). For the second component (referred to as the *choice of display* component), the display to attend to was learned in terms of reducing the effort/time of moving attention from one display to the next. This component was only utilized when SVS was available because it was only with SVS that a choice of displays for redundantly available information (e.g., altitude available on both the PFD and SVS) was applicable. The reduction in effort/time resulted from an increased likelihood that the next display item needed would be available on the currently-attended display.

Although the full details of learning in the model are outside the scope of this report, some key equations are presented here to offer some insight into learning in ACT-R. Recall from section 5.3 that the production utility is:

$$U_i = P_i G - C_i + \varepsilon$$

where P_i is the probability of success that the production rule i, if executed, can achieve the current goal; G is the value of the goal; C_i is the cost (typically the time required to perform an action); and ε is a noise parameter.

When production learning is enabled using the decay form of the equation, the probability of success and cost of a production utility represent past history with an emphasis on what has occurred more recently as specified by:

$$P_i = \frac{\sum_{n}^{successes} t_n^{-d}}{\sum_{n}^{successes} t_n^{-d} + \sum_{m}^{failures} t_m^{-d}}$$

Probability of success equation with decay

$$C_i = \frac{\sum_{k}^{(successes+failures)} t_k^{-d} Cost_k}{\sum_{n}^{successes} t_n^{-d} + \sum_{m}^{failures} t_m^{-d}}$$

Cost equation with decay

where *successes* is the number of successes; *n* represents each success; *failures* is number of failures; *m* represents each failure; *k* represents each success and failure; $Cost_k$ is the time/effort to succeed or fail for each success and failure; t_n, t_m, t_k represent the time elapsed since each success, each failure, or each success/failure occurred, respectively; and *d* is a decay parameter.

In the MA&D/CMU model, choosing a display and choosing a display item on that display were separate ACT-R goals, requiring separate sets of production rules and utilities. For the *information need* component, P_i corresponded to the rewarding information discussed above whereas C_i was set to a constant. For the *choice of display* component, P_i was set to 1 (meaning the goal of moving to or remaining at a display was always successful) whereas C_i corresponded to the time/effort of moving attention to a display.

How the *choice of display* component enabled learning in the MA&D/CMU model has already been discussed in section 5.5. The *information need* component was more complex in its implementation and is presented here. Recall that multiple production rules can satisfy their respective conditions, but only one of those, the one with the highest utility, can execute its corresponding action. In the MA&D/CMU model, multiple production rules were created to determine the information needs of the pilot. Each production rule's action corresponded to a different display item for the pilot to attend to – one specified altitude, another specified air speed, and so on. However, the success of this production rule was not simply that the next display item was successfully attended to, but rather that piece had changed in value by some amount as specified by the modeler. Further success was assigned if the changed value then triggered a reactive/procedural task. Thus, the learning of the top-level goal (i.e., move attention to another display item) took the success and failure of its sub-goals into account when determining its utility by propagating success and failure back up the goal stack.

As mentioned in section 4.2, to accomplish this with the standard version of ACT-R 5.0 was not possible. Instead, many aspects of the ACT-R 4.0 hierarchical reinforcement scheme, which was based on the goal stack from ACT-R 4.0, were implemented in a variant version of ACT-R 5.0. Despite this added complexity, learning based on propagation up the goal stack works essentially the same as single production learning. For example, each of the production utilities for the *information need* component (e.g., altitude, air speed, etc.) was initialized identically at the

beginning of the run in terms of success (i.e., no successes or failures yet recorded). As such, the highest utility was based solely on the stochastic effect from the noise term. Thus, the pieces of information selected for the pilot to attend to early on were purely random. Over time, the successes and failures of the sub-goals (was the information changing and did it trigger pilot actions?) became more important in terms of guiding the selection towards rewarding information. However, because these decayed over time, the recent successes and failures carried more weight. Indeed, the MA&D/CMU team believed it was critical to use the decay form of the probability of success and cost equations because this enabled rapid adaptation to changing information needs.

5.10 Resultant and emergent behavior

This section describes the capabilities of the modeling tools for demonstrating resultant and emergent human behavior. Resultant and emergent behaviors are often confused in this context and the difference between the two is worth mentioning. Resultant behavior is that which could be "reasonably expected" from a collection of objects and the interactions between them based on an understanding of their individual behavior (Auyang, 1999). In contrast, emergent behavior is that which is truly “unexpected”. For both the instrument approach modeling and taxiway operations efforts, it is the opinion of the authors that the behavior predicted by the HPM tools demonstrates resultant rather than emergent behavior. In most HPM efforts, by the time the modeling teams have constructed their models, understood the implications of the underlying assumptions, and analyzed the results, the behavior predicted should be reasonably expected.

For the taxiway operations modeling effort, the resultant behaviors predicted by all the teams’ models were types of taxiway errors (wrong turns or missed turns) made by the simulated flight crew. The different team approaches to modeling these errors is discussed in section 5.8. In contrast, for the instrument approach modeling effort, the teams had more leeway in choosing problems to study based on their interest and expertise (e.g., emphasis on workload vs. visual attention allocation). Thus, the resultant behaviors predicted by the modeling tools are more varied and are discussed further below.

ACT-R Rice/UIUC

The key resultant behavior of the ACT-R Rice/UIUC model was visual attention allocation. The key metric was the percentage that the pilot fixated on each of the displays (referred to here as *fixation percentage*). In addition, the model predicted the transition of visual attention from one display to display to another (e.g., display A to display B, display B to display A). The metric for this was a transition matrix that showed the percentage amount.

Another resultant behavior was the decision made by the pilot to go-around or land based on the aircraft’s alignment, amount of offset, approach angle, and personal attributes of the pilot such as risk-averse nature.

ACT-R MA&D/CMU

The key resultant behavior of the ACT-R MA&D/CMU model was visual attention allocation with an emphasis placed on capturing the behavior of pilots who chose to bias their attention allocation towards either the SVS or traditional displays. The key metric was dwell percentage. Another resultant behavior was the decision made by the pilot to go-around or land based on the aircraft’s offset from the runway and the pilot’s ability to see the runway at decision altitude.

Air MIDAS

Workload was the key resultant behavior predicted by Air MIDAS. (Workload as a metric is discussed in section 5.6.) Other resultant behaviors include pilot response times in reaction to key

events such as “runway in sight”, “go around due to an ATC instruction”, and “go around due to inability to see the runway at decision altitude”.

Lastly, visual attention allocation was also resultant behavior of Air MIDAS, but recall that background scan pattern, which is the dominant contribution to overall allocation, was an input to the model rather than an output (see section 5.5). Thus, the results to a large extent reflect the behavior of the subject pilots in the Santa Barbara part-task simulation. The metric for visual attention allocation was fixation percentage, dwell duration measured in seconds, and dwell percentage.

D-OMAR

The key resultant behavior of the D-OMAR model was visual attention allocation. Another resultant behavior was the decision made by the pilot to go-around or land based on the aircraft’s offset from the runway, distance to the runway threshold, and a personal attribute of the pilot related to his/her priority to land.

A-SA

Resultant behavior of A-SA comes in two forms. The first is the probabilities that the pilot will visually attend to displays and the out-the-window view (via the SEEV model). The second is error predictions at specific decision points based on the pilot’s SA, where the pilot’s SA can vary between 0 – 1 (1 being perfect SA) as a function of time and events.

6 Verification and validation

The ability to accurately model operator performance requires verification and validation, a tedious, but necessary process in HPM development. During the **verification** phase, the complete HPM software package is (or should be) tested to establish that it performs as intended by the modeler. Verification can be an exhaustive process in which all aspects of the model must demonstrate that they are implemented correctly. For example, verification should demonstrate that the scheduler schedules correctly, the random number generator produces truly random numbers, and the communication link between the HPM and the external environment works as designed. In the strictest sense, verification means the model is bug-free. In reality, there may be bugs in the model that do not affect the performance or output of the model to the extent that they can be detected through methodical analysis. Proof of verification is usually not explicitly required in a modeling effort such as the NASA HPM Element, but rather assumed as a normal part of the model development and analysis process. However, sometimes lack of verification is evident when model results are counter-intuitive.

The **validation** phase focuses on the ability of the model to provide sound predictions within certain bounds. Of course, "sound predictions" and "certain bounds" are subjective. The bounds could include phase of flight, weather conditions, flight deck configuration, or types of procedures. The classic approach to validation (often referred to as *predictive validation*) is to model some baseline scenario in which some type of human performance data is available (e.g., fixations percentages) for comparison. If the differences between the model predictions and the baseline data are unacceptable, the model is tuned/modified as needed (often referred to as calibration), re-run, and compared against the data once again. This process is repeated until the comparison between predictions and data are deemed acceptable. The model is then run for another scenario/condition where data is available, but has not been previously utilized for comparison. If the comparison between predictions and the new data are acceptable, the model is considered validated. At this point, the validated model can be used to make predictions within certain bounds.

6.1 Validation of specific models

This section discusses validation of the individual HPM efforts based on results presented in their respective final reports. It should be noted that validation based on only 3 subject pilots should be treated with care since HPM tends to focus on average human performance whereas with only 3 subjects it would be statistically feasible to have all pilots either perform above or below the average. Thus, the modeler may assume that there is a problem with the model when in fact the model may be behaving quite well.

The flight segments and scenarios are defined as follows:

Flight segment 1. Start to Initial Approach Fix (IAF)
Flight segment 2. Initial Approach Fix (IAF) to Final Approach Fix (FAF)
Flight segment 3. Final Approach Fix (FAF) to Decision Height (DH)
Flight segment 4. Decision Height (DH) to End

Scenario Number	Description	Outcome
1	Baseline VMC	Disengage autopilot and prepare to land
2	Baseline VMC	Late reassignment (land on parallel runway)
3	Baseline VMC	Terrain mismatch (traffic on runway)
4	Baseline IMC	Nominal landing (land on parallel runway)

5	Baseline IMC	Missed approach (go around)
6	Baseline IMC	Terrain mismatch (go around)
7	SVS IMC	Disengage autopilot and prepare to land
8	SVS IMC	Late reassignment
9	SVS IMC	Missed approach (go around)
10	SVS IMC	Terrain mismatch (go around)

ACT-R Rice/UIUC

The key resultant behavior of the ACT-R Rice/UIUC model was visual attention allocation and the key metric was the percentage that the pilot fixated on each of the displays (referred to here as *fixation percentage*). The SVS condition of flight segment 2 was used for calibration resulting in an r-squared fit of 98%. After calibration, the fixation percentages were predicted for three flight segments and compared against the Santa Barbara data. The resulting r-squared fit explained about 85% (baseline flight segment 1), 68% (SVS flight segment 1), and 93% (baseline flight segment 2) of the variance, respectively.

In addition, the model predicted the transition of visual attention from one display to display to another. The metric for this was a transition matrix that showed the percentage of transitions from one display to another occurred. The resulting r-squared fit explained 79% (baseline segment 1), 42% (SVS segment 1), 77% (baseline segment 2), and 69% (SVS segment 2) of the variance.

ACT-R MA&D/CMU

The key resultant behavior of the ACT-R MA&D/CMU model was visual attention allocation and the key metric was dwell percentage. The model was executed for approximately 280 seconds, representing the latter half of the approach (vs. approximately 600 seconds for the subject pilot runs). Comparing the model to subject 3 across all scenarios, the r-squared fit explained 69% of the variance. Comparing the model to subject 4 across all scenarios, the r-squared fit explained 82% of the variance. Comparing the model to subject 5 across all scenarios, the r-squared fit explained 86% of the variance. Aggregating subjects 3-5, the r-squared fit explained 85% of the variance.

Air MIDAS

During the initial phase of the SVS study (2003), validation of visual attention was performed. The key metric was fixation percentage. The r-squared fit for Scenario 4 explained 58% of the variance (r = 0.7608). The r-squared fit for Scenario 7 (SVS) explained 77% of the variance (r = 0. 8782). The r-squared fit for Scenario 8 (SVS with sidestep) explained 31% of the variance (r = 0.5538).

D-OMAR

The key resultant behavior of the D-OMAR model was visual attention allocation and the key metric was dwell percentage. Although no statistics were provided, the model predictions look reasonable when compared to the three subject pilots with the exception of the out-the-window view, which the model over-predicts.

Human Subjects and Model Eye-tracking Data: IMC Nominal Approach - IAF to FAF

Condition	IMC				SVS			
	Subject 3	**Subject 4**	**Subject 5**	**Model**	**Subject 3**	**Subject 4**	**Subject 5**	**Model**
off	2.24	3.90	5.15		2.40	4.76	1.53	
OTW	0	1.00	0.35	6.31	0	4.82	0.06	6.15
SVS					24.02	15.25	11.84	26.03
PFD	43.63	55.95	34.42	37.00	27.93	31.36	42.57	26.11
NAV	32.62	33.52	54.71	47.44	28.96	39.55	37.39	34.03
other	21.51	5.49	5.36		16.69	4.26	6.62	

Human Subjects and Model Eye-tracking Data: IMC Nominal Approach - FAF to DH

Condition	IMC				SVS			
	Subject 3	**Subject 4**	**Subject 5**	**Model**	**Subject 3**	**Subject 4**	**Subject 5**	**Model**
off	2.22	4.08	2.84		1.80	5.07	1.53	
OTW	0	11.57	2.03	10.39	0.52	4.37	2.57	8.55
SVS					19.22	33.53	49.04	28.37
PFD	39.98	46.72	42.54	39.60	41.21	35.96	24.76	29.53
NAV	33.45	32.40	44.64	39.62	26.90	19.53	18.16	28.66
other	24.35	5.22	7.94		10.36	1.55	3.95	

A-SA

The SEEV model of A-SA estimates the visual attention allocation based on coefficient estimates (see the report for more details). The A-SA team provided significant detail into the validation effort in their Appendix from the 2003 report. The report discusses results in terms of correlation r rather than goodness of fit r-squared. In order to compare with the other validation results presented, the results were converted to r-squared. In general, the model under-predicted the ND and out-the-window allocations. Using the Santa Barbara data across scenarios 6-10, the resulting r-squared fit explained between 36% and 64% of the variance. The A-SA team performed validation against a richer SVS data set collected at UIUC and the resulting r-squared fit explained between 70% and 86% of the variance.

7 Usefulness to the design and evaluation of new technology

Identify the questions to be answered

The usefulness of the HPM tools to the design and evaluation of new technology is only partly related to existing core capabilities provided by the tools. Equally as important to the design and evaluation process is the need to identify the key questions being asked that the modelers and modeling tools are trying to answer. For example, if the NASA HPM Element had required all teams to predict workload as an output of their models then one can assume the modeling teams would have made the necessary model modifications to do so (only Air MIDAS actually had this capability). For future modeling efforts, the question being asked will ultimately determine what a model is capable of predicting. Modelers will focus their effort and expend the resources to ensure that the question being asked will be answered. It is useful to keep in mind that HPM does a better job of predicting relative measures rather than absolute measures. Thus, the question, “Is cockpit A better than cockpit B?” is easier to answer than “Is cockpit A ready for full-scale prototype development?” Future HPM customers must be careful to ask the key questions prior to

any significant model development to ensure the key answers can be realized. If the HPM customer leaves the questions more open to the modelers for exploration, it is critical that the modelers articulate the issues they have chosen to focus on, their reasons for choosing those issues, and the specific results achieved relative to that focus.

Knowing the customer

Understanding the customer is always an important aspect of providing a product or service and this holds true with HPM as well. One aspect of knowing the customer is to speak in a common language. Aviation managers with engineering backgrounds generally do not understand the language of cognitive science. Thus, for example, much of the language used in this paper to describe the models would likely lead to customer confusion and frustration. Where possible, it is important to think and talk in terms of the customer's metrics and constraints since the customer is more likely to gravitate to the modeler and modeling tools that s/he understands best.

Understanding HPM tool capabilities

The architecture of the modeling tools does have bearing on how readily core capabilities such as workload, visual attention allocation, crew interactions, procedures, situation awareness, and error prediction can be modeled and predicted. For example, multiple operator models (e.g., pilot-flying and pilot-not-flying) are more easily accommodated by Air MIDAS and D-OMAR. Thus, if flight crew or pilot/ATC interactions are expected to be significant drivers to a future modeling effort then the Air MIDAS and D-OMAR tools would be more straightforward to apply. In contrast, the A-SA and ACT-R models focused specifically on what drives visual attention from a bottom-up (e.g., effort to move the eyes) as well as top-down perspective. Hence, if visual attention allocation needs to be understood for a particular technology, the ACT-R and A-SA modeling tools would more easily facilitate the analysis. Next, as was mentioned above, only the Air MIDAS model provides an explicit workload metric as an output of the model. However, the D-OMAR and two ACT-R models have workload constraints (via human and system resource limitations) embedded in their models so metrics reflecting the underlying resource utilization information necessary could be output from their models with some modification. Multi-tasking is explicitly supported by the frameworks of Air MIDAS and D-OMAR. Lastly, complex memory modeling is available in ACT-R, which can be useful in modeling human errors related to memory loss.

Each of the HPM tools discussed in this paper has extended their capabilities significantly to support the instrument approach and taxiway operations modeling efforts. An evaluation of new aviation technology beyond these efforts would most likely require modifications to the models to account for the specific details of the new technology, associated procedures, and perhaps different phases of flight such as departure or cruise. However, this would be a smaller effort compared to the work already expended during this HPM effort for the development of the external models to represent the aircraft flight dynamics, flight deck displays, and the communication link between external environment and HPM tool (recall that the teams expended between 25-70% of their effort to represent these functions). The teams that connected to higher fidelity flight simulators (the ACT-R Rice/UIUC and Air MIDAS teams) should be acknowledged as they are poised to tackle problems in which the closed-loop behavior between the pilot's action and the aircraft's response (and vice versa) is a key factor.

Of course, the respective modeling tools and capabilities are dynamic. Each successive modeling effort in a complex environment such as aviation most likely adds to a tool's repertoire of capabilities. HPM has come a long way, but still has a long way to go as well. It is likely that in two years the HPM tools discussed in this document will be able to further demonstrate enhancements applicable to aviation modeling.

HPM support: Determining information needs

An important aspect of future HPM is the ability to provide modelers with information needs of the pilots, which also includes the frequency that the information is needed. In the instrument approach effort, the information needs were obtained through a task analysis, but the frequency that this information needs to be visually attended to was unknown. For most of the modeling teams, the frequency of the information needs was essentially obtained by working backwards from the eye-tracking data – either directly or through the model calibration step. However, collecting eye-tracking data and then post-processing it is difficult and expensive. Since one of the stated goals of HPM is to reduce the need for human-in-the-loop simulations, future task analyses efforts to gather data from SMEs should focus specifically on the complete set of information needs (and their frequency of update) to maintain situation awareness. On the other hand, learning (as demonstrated by the ACT-R/MA&D team) could be quite useful in this regard as could be A-SA. A-SA strives to address the same issue, but in a more prescribed manner, perhaps future efforts should consider acquiring information specifically identified by the A-SA framework. Indeed, a worthwhile objective to pursue would be to incorporate the A-SA framework into the other modeling tools.

Lesson learned: Importance of stating assumptions

A key lesson learned specifically related to conducting the cross-model comparison is the need for the modeling teams to state all their assumptions explicitly for future modeling efforts. For example, ACT-R has several mechanisms for modeling memory, but for any given modeling effort some of these would be disabled. The ACT-R MA&D/CMU team disabled spreading activation whereas the ACT-R Rice/UIUC team disabled partial matching, but in neither case were these identified in their final reports. Thus, if someone is familiar with the memory modeling in ACT-R, it can be quite perplexing to understand how domain-specific memory is incorporated without knowing these assumptions. As not to single out ACT-R, there were several more examples of unclear assumptions that involved the other modeling teams as well. It should be noted that the Air MIDAS team did provide a useful appendix for many of the model assumptions that perhaps could serve as a template for the other HPM tools for future efforts. The authors firmly believe that understanding the underlying assumptions is fundamental to understanding the usefulness of the HPM tools to the design and evaluation of new technology.

8 Acronyms

ACT-R	Adaptive Control of Thought - Rational
Air MIDAS	Air Man-machine Integrated Design and Analysis System
A-SA	Attention – Situation Awareness
ATC	Air Traffic Control
AvSP	Aviation Safety Program
ConOps	Concept of Operations
D-OMAR	Distributed Operator Model Architecture
FY	Fiscal Year
HPM	Human Performance Modeling
IMC	Instrument Meteorological Conditions
IMPRINT	Improved Performance Research Integration Tool
ND	Navigation Display [aka Horizontal Situation Indicator (HSI)]
PFD	Primary Flight Display
RVR	Runway visual range
SA	Situation Awareness
SEEV	Salient, Effort, Expectancy, Value
SME	Subject Matter Expert
SVS	Synthetic Vision System
SWAP	System-Wide Accident Prevention
UDP	User Datagram Protocol
VMC	Visual Meteorological Conditions

9 References

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y., (2004). An Integrated Theory of the Mind. http://act-r.psy.cmu.edu/papers/403/IntegratedTheory.pdf

Auyang, S.Y., (1999). Foundations of Complex-system Theories: In Economics, Evolutionary Biology, and Statistical Physics, Cambridge University Press.

Card, S.K., Moran, T.P., Newell, A. (1983). The psychology of human computer interaction. Hillsdale, NJ: Lawrence Erlbaum.

Deutsch, S., (1998). Multi-disciplinary foundations of multiple-task human performance modeling in OMAR in the *Proceedings of the Twentieth Annual Meeting of the Cognitive Science Society*, 1998.

Endsley, M. R. (1988). Design and evaluation for situation awareness enhancement. *In Proceedings of the Human Factors Society 32nd Annual Meeting* (pp. 97-101). Santa Monica, CA: Human Factors Society.

Goodman, A. (2001). Enhanced descriptions of off-route navigation errors in T-NASA2, NASA Ames Research Center.

Hooey, B. L., Foyle, D. C., & Andre, A. D. (2000). Integration of cockpit displays for surface operations: The final stage of a human-centered design approach. SAE Transactions: Journal of Aerospace, 109, 1053-1065. http://human-factors.arc.nasa.gov/ihi/hcsl/publications.html

Goodman, A., Hooey, B.L., Foyle, D.C., & Wilson, J.R., (2003). Characterizing Visual Performance During Approach and Landing With and Without a Synthetic Vision Display: A Part-Task Study in *Proceedings of the 2003 NASA Aviation Safety Program Conference on Human Performance Modeling of Approach and Landing with Augmented Displays*. http://human-factors.arc.nasa.gov/ihi/hcsl/publications.html

Keller, J., Leiden, K., & R., S. (2003). Cognitive task analysis of commercial jet aircraft pilots during instrument approaches for baseline and synthetic vision displays in *Proceedings of the 2003 NASA Aviation Safety Program Conference on Human Performance Modeling of Approach and Landing with Augmented Displays*. http://human-factors.arc.nasa.gov/ihi/hcsl/publications.html

Leiden, K., Laughery, K.R., Keller, J. W., French, J.W., Warwick, W. & Wood, S.D. (2001). A Review of Human Performance Models for the Prediction of Human Error. http://human-factors.arc.nasa.gov/ihi/hcsl/publications.html

McCracken, J.H., & Aldrich, T.B., (1984). Analyses of selected LHX mission functions: Implications for operator workload and system automation goals (Technical Note ASI479-024-84), Fort Rucker, AL: Army Research Institute, Aviation Research and Development Activity.

Pashler, H. (2000). Task switching and multitask performance. In Monsell, S., and Driver, J. (editors). Attention and Performance XVIII: Control of mental processes. Cambridge, MA: MIT Press.