Technical Research in Advanced Air Transportation Concepts & Technologies (AATT)

Task Order (TO) 69
ATM Human Behavior Modeling Approach Study

Ben P. Wise
Mary McDonald
Lisa M. Reuss
Jesse Aronson

November 14, 2001

Science Applications International Corporation (SAIC)
1100 N. Glebe Road, Suite 1100, Arlington, VA 22201

Operating Under Contract NAS2-98002

National Air and Space Administration (NASA)

Table of Contents

Figures

Tables

Appendices

1 Executive Summary

The Aviation System Capacity (ASC) Office at NASA Ames is engaged in several activities that require non-real-time modeling of human behavior and automation in ATM. These activities are supported by the AATT Project and by the Quiet Aircraft Technology (QAT) Program. Within the AATT / Benefits and Safety Assessments (B&SA) Element, the M&S Task Area seeks to provide the high-fidelity modeling capability required to assess new tools or concepts for the NAS. The objective for M&S is to provide the ability to perform multi-objective assessments of Distributed Air-Ground (DAG) concepts. Within the QAT / Community Noise Impact Project, the Noise Mitigation Controller Tools (NMCT) Element has an objective of demonstrating, via laboratory simulation, the effectiveness of a controller decision support tool (DST) for low noise approach and departure procedures. Each of these activities will utilize components of an integrated suite of ATM modeling tools.

The modeling capabilities developed within AATT and QAT will also serve as the basis for future NAS simulation environments such as those proposed within the Aviation System Technology Advanced Research (AvSTAR) vision. Modeling tools developed will be used to conduct system-level assessments of advanced ATM concepts. Thus, the M&S framework must be flexible to promote re-use for a broad range of NAS concept evaluations and/or trade studies.

All of the activities described require non-real-time modeling of human behavior and modeling of human behavior and automation was in the early stages of development. Further, given the immense importance of human factors/automation in the design and operation of any proposed advanced ATM concept, this category of modeling was found to require urgent attention. For AvSTAR assessments, NASA also sees a requirement to develop human "team modeling" capabilities. The first steps in building such a capability are to determine the human constraints on the ATM system, to assess the state-of-the-art in human behavior modeling, and to formulate an approach for developing human behavior models, which can be integrated with the ATM modeling tool suite.

The overall purpose of this study is to provide NASA with the background information required to identify key elements of human behavior automation modeling and an approach for a phased implementation of these elements into a non-real-time modeling environment, which includes humans and automation in ATM.

This report is broken into four major sections and five appendices:

This first section gives background and study requirement information.

The second section describes the components involved in the representation of human behavior in the ATM system.

The third section describes our assessment of the state-of-the-art in human behavioral modeling. It includes the evaluation of simulations and programs inside as well as outside the ATM realm such as in industry, academia, and the military. We want to ensure that NASA is aware of the evaluated "best of the breed" both inside and outside of its resident domain. This third section also includes our assessment of the simulations and programs evaluated. Finally, in section three, there is a brief description of some relevant technical "tech" feeder programs occurring in the modeling and simulation world that we believe stand to impact the way the NAS system in modeled and analyzed.

Section four describes and illustrates our recommendation for a roadmap to integrate existing models and programs into an ATM Modeling Environment. We describe our "toolkit" approach with associated features and issues, and describe what the "pieces" are. Also included is a rough timeline for accomplishing the proposed integration.

In addition to the four report sections, we have also provided five appendices.

Appendix A contains a list of acronyms that appear in this report.

Appendix B contains a list of references used by the authors in researching and compiling this report.

Appendix C describes two "use cases", two examples of the human behavior matrix presented in section two filled out in lower and higher fidelity formats, to illustrate the practical difference in changing the level of fidelity of a model of human behavior.

Appendix D contains more detail on issues involved in the modeling of human behavior. Again, it is a rich and complex field, and the purpose of the appendix is to begin to give a reader appreciation for the magnitude of choices/details when deciding on a particular representation of human behavior for a given model in a given domain.

Lastly, Appendix E contains a brief description of general techniques for modeling human behavior. Although the focus of this study was not to provide an academic discussion of the specific computational or mathematical techniques being employed in the specific simulations and programs, we thought it would be useful to NASA Ames to be able to recognize key terms and concepts with regard to some of the various ways that human behavior is modeled. We have provided a brief description of these methods, along with a fairly simplified view of each method's pros and cons. It is important to understand that the modeling of human behavior is a task of extreme complexity and magnitude, and that there is, simply, "more than one way to skin a cat." One of the reasons why the field is so rich with research is because there are so many different ways to approach the challenge of modeling human behavior.

With this report, NASA should be able to understand what areas require more research as well as what areas do not apply to future ATM modeling and therefore no further investigation.

1.1 Background

The representation of human behavior is a rich and complex topic. As a definition of human behavior representation (HBR), we will use the one offered by Pew and Mavor in *Modeling Human and Organizational Behavior*, that is, a "computer-based model that mimics either the behavior of a single human or the collective action of a team of humans."(reference 13)

There are many ways of viewing the challenge of representing human behavior. For instance, one's goal may simply be to simulate/emulate the effects of a human decision, without providing much detail on the cognitive events, which take place that give rise to that decision. In other cases, specifically modeling the cognitive processes may take center stage. In yet other cases, modeling human performance (e.g. visual and auditory perception, motor skills, etc) may be what is of most interest.

Before any attempt can be made to represent the human behavior resident in any domain, some thought must be given to how to frame the problem. More specifically, one can view the ultimate representation as a set of people, functional behaviors, and processes, or one may choose to view the representation as a set of flows and controls. Each framework gives rise to unique modeling considerations and requirements, and it is our view that one should take multiple approaches in specifying the human behavior before any attempts are made to "code" it in a model. In the end, choices will have to be made, so as not to give rise to an inordinately complex model. There will be overlaps and redundancies in taking the time to approach the problem from more than one framework, but it will also help to ensure that key components of the desired human behavior are not missed.

For example, if one chooses the first framework, the knowledge engineer (KE) responsible for creating the model should try to address (at a minimum) the following questions with a subject matter expert:

- Who are the people/actors in this system?
- What are their primary behaviors (or functions)?
- What are the underlying processes involved in the generation of these behaviors (e.g. communication, motivation, information processing, etc)?

In contrast, if the latter framework is chosen, the KE should try to address (at a minimum) the following questions:

- What entities are moving from place to place?
- What are the "roadways" and /or other limitations to movement?
- Who is directing or controlling flow?
- For those directing or controlling, what are the cues or measures they pay attention to?
- If the physical analogy of a queue is appropriate, what is the capacity of that queue?

- What determines how long does an entity stays at one location before moving to the next?
- Is the process discrete or continuous?

There is also the issue of level of resolution desired in a specific representation of human behavior. In a nutshell, the level of resolution of a model or simulation describes its level of detail in representing some aspect or aspects of human behavior. The higher the level of resolution, the more detail which is contained, and *generally* the larger and slower running the model. There is no set one answer or way to resolve this issue. There are pros and cons to increasing or decreasing the level of fidelity of a given representation of human behavior. How much fidelity is "required" for a specific analysis is situation and question dependent. It is becoming more common in the modeling and simulation (M&S) world to allow for multiple levels of resolution, either within one model, or by allowing a way to transition smoothly between models of different levels of resolution. When we discuss relevant M&S technical or "tech" feeder programs, we will discuss techniques and methods, which may be of use in an ultimate modeling toolkit of human behavior in the Air Traffic Management (ATM) realm.

1.2 Requirements of this Study

SAIC shall participate in a kick-off meeting at NASA Ames to develop an understanding of the automation tools, concepts and procedures that may be evaluated in ATM concept simulations.

Task 1: <u>Identification of Human Components in ATM System Modeling</u>
SAIC shall identify and document the human roles and behaviors that must be represented within a system-wide model of the NAS.

Task 2: <u>Assessment of State-of-the-Art in ATM Human Behavior Modeling</u>
SAIC shall survey existing ATM human behavior/automation models to identify those models that can be applied within a non-real-time modeling environment for conceptual evaluation of ATM tools, procedures and concepts. This survey shall extend the previous survey conducted for AATT (reference 6) by providing newer and/or updated information in the particular domain of human behavior modeling. The survey shall identify both strengths and limitations of the models including their known range of applicability, ease of use, availability to NASA, and computational requirements. Any human behavior models that could also be used within a real-time simulation environment should be noted. The survey shall also identify human behavior models from other fields [e.g. models used by Defense Advanced Research Project Agency (DARPA), Department of Defense (DOD), Department of Transportation (DOT), and other agencies] that could be adapted for ATM applications. The survey should include a discussion of the simulation frameworks that are used in conjunction with current human behavior models.

Task 3: Roadmap for Integrating Human Behavior into an ATM Modeling Environment. Based on the results for Tasks 1 and 2, SAIC shall develop a phased implementation plan for automating human behavior within a non-real-time, ATM modeling and simulation environment.

1.3 Modeling Assumptions

In any discussion about the modeling and/or simulation of a domain, one must first address which assumptions, if any, are to be considered. These assumptions will help guide the development of the M&S towards maximum utility and usefulness, while at the same time, assuring that the actors and functions in the given domain are modeled as accurately and as efficiently as possible.

SAIC expresses its thanks to Ms. Sandy Lozito of NASA Ames, for supplying the following modeling assumptions. It was important to us to record/address the specific assumptions that NASA Ames believes are appropriate to the modeling of the National Air Space (NAS) domain. We understand that the list of assumptions is a dynamic list, growing and responding to new realities in the domain. As a case in point, the incidents in the United States (U.S.) on September 11, 2001 certainly will have an effect on the first modeling assumption listed below, namely if the number of aircraft in the NAS will in fact increase as quickly as once thought. In any case, SAIC wishes to extend their deepest sympathies to all who were involved or impacted by the tragic events, but at the same time, expresses their confidence in the industry and in the eventual truth of the first assumption listed below.

- Number of aircraft in NAS is rapidly increasing (thus, emphasize capacity)
- All three parts of triad will be involved in these changes (Air traffic controller, ground controller, and pilot)
- Demographics
- Some shift in roles and responsibility, including dynamic shifts
- Changing and dynamic airspace structure (e.g., dynamic resectorization)
- More flexibility for the users
- Better weather prediction and distribution of weather information
- Better data and use of Special Use Airspace (SUA)
- More aircraft intent will be available
- Intermodal interface considerations
- Considerations for the Small Aircraft Transportation System (SATS)
- More data link communications (air-air and air-ground)
- Air to air data sharing available
- Use of new ground and airborne automation tools (e.g., conflict prediction & resolution)
    - Ground conflict probe
    - Airborne (Cockpit Display of Traffic Information (CDTI)) and probe
    - Trajectory negotiation between air and ground.

In addition to specifying model assumptions, it is also helpful to consider what the overall process is involved in specifying modeling scenarios. Again, SAIC wishes to thank Ms. Sandy Lozito for supplying her priorities. The process is described below.

- Consider the research question
- Consider the technical and logistic constraints (e.g., computing power, staffing)
- Decide which facets of the research questions can be addressed in a particular scenario
- Use background research to further refine those questions
- Make basic decisions for scenario development (e.g., traffic density, number of operators)
- Consider communications requirements (if any)
- Begin development of the scenarios
- Conduct any early evaluation of the scenarios with users
- Check data collection
- Make appropriate modifications, refine scenarios.

````
````

2 Identification of Human Components in ATM System Modeling

2.1 Top Level Purpose

As a key element in analyzing human behavior models we have created a matrix to represent different types of human decision-making. Use of such a matrix allows categorization of the decisions, which need to be modeled in the NAS domain. It also provides a uniform mechanism by which to evaluate different human behavior modeling systems and techniques. This section describes the matrix and its use.

2.2 What is Human Behavior Representation?

Human Behavior Representation (HBR) again, is, at its most succinct, the representation of the decision-making processes and decision-related actions of humans represented within a simulation. HBR can cover a broad range of behaviors, from the perceptions and actions of an individual such as a single air traffic controller up to the collective behavior of an entire command and control system, such as the entire NAS. HBR also represents a breadth of fidelities, from simple models which make entities appear to be behaving correctly when viewed from afar, to detailed models of human sensory and cognition processes.

HBR is closely related to workload modeling, human factors modeling, and understanding human behavior processes. It is important to emphasize that each one of these techniques has its own important uses, its own community of experts, and its own challenging problems. However, our task is to focus on computational models that can represent, or emulate, human behavior relevant to the future-NAS analysis problem.

HBR is different from workload modeling in that the emphasis is on producing realistic behaviors, not on measuring the effort it would take for humans to perform. This differs from human factors modeling in that HBR usually does not attempt to measure effects such as decreasing performance with fatigue, or the factors that influence the rate of fatigue or recovery there from (though it may represent such factors as part of a model).

HBR varies from understanding how human behavior works because the emphasis is on producing realistic behavior, by computational means. The computation models used need to mirror the results of human decision-making, but may not match the actual cognitive processes used by human. If people were performing a task by an unknown process, then an accurate description of human behavior must include that task. However, we could not produce a computationally realizable representation of such human behavior. On the other hand, if linear programming (LP) produced a good approximation to the output of a person’s planning task – regardless of whether or not we can describe how they actually did the task - then LP is likely a good enough representation of the behavior, even though people clearly never execute LP in their heads.

Our focus, therefore, is on computational techniques for simulating the mission-relevant externally observable behavior of individuals or groups of people.

Many engineers automatically assume that modeling for HBR is limited to using numerically based techniques, such as control theory or stochastic dynamic programming. We do not. Many computer science or artificial intelligence techniques operate exclusively with symbolic, non-numerical data (think of a compiler, that takes in computer programs in a high level language, and outputs equivalent programs in a much simpler language), and we will feel free to examine any such techniques that seem applicable.

Many engineers and computer scientists also automatically assume that respectable computational procedures are deterministic, and that the role of random numbers is limited to "random number seeds" in repeated "Monte Carlo" simulation runs. We do not. Randomized algorithms are sometimes the most efficient known approaches even to "hard-core" mathematical problems (think of the simple Miller-Rabin procedure to test an integer for being prime or composite, which is many orders of magnitude faster than the best known deterministic algorithm). Randomized algorithms are also a standard way of avoiding "threshold effects", or artifacts of behavior that occur due to repeatedly accessing the same conditions.

2.3 Behavioral Matrix

In order to characterize and assess both the required elements of human behavior modeling and the capabilities of existing tools and techniques, we developed a HBR taxonomy matrix. This matrix is a top-level orientation tool, suitable for multiple diverse domains - military, NAS, or economic modeling. We start with a very abstract framework that is oriented toward computable representations of human behavior, and use it as an organizing framework to categorize our knowledge about the NAS and related HBR, and to surface issues in future modeling. Figure 2.3-1 shows the form of the matrix. Any behavior representation requirement or approach occupies some portion of the two-dimensional (2D) space described by the matrix. Consequently, the matrix identifies taxonomies by identifying HBR elements, which cluster in the same regions of the matrix.

	Current Situation Perception	Future Situation Projection	Option Generation	Outcome Evaluation
Standard Reactions				
Routine Performance				
Staff Work				
Judgement Calls, Problem Solving				

Figure 2.3-1 – Top Level HBR Taxonomy

The HBR taxonomy matrix contains four rows and four columns. The rows of the matrix represent the depth and flexibility of the behaviors, while the columns represent the temporal dimension of the HBR model – from reacting to current situations to anticipating and planning for future situations. The four rows are roughly hierarchical, where each is situated in the context from the next lower, more fundamental level. The top two layers are commonly described as “canned behaviors”, while the lower two layers represent more dynamic, less scripted and therefore more “human-like” behavior.

2.3.1 Standard Reactions

The topmost row represents canned short-term responses to short term contingencies. Examples from the NAS include the following: a pilot will immediately abort the landing if an obstacle is seen on the runaway, and a pilot will immediately take evasive action at altitude if a midair collision suddenly appears imminent. No thought is required as to how to manipulate the controls; no debate as to proper course of action is permissible. They take action immediately, and they do it roughly the same way every time. Similarly in the military domain, if a line formation of tanks traveling down a road is attacked from the air, they will immediately begin firing back and simultaneously move into a herringbone formation. No discussion, no problem solving, no orders.

Standard reactions can be modeled by a number of techniques including procedural code, rules and finite state machines (FSMs) (see the following section).

2.3.2 Routine Performance

The next layer contains the standard, long-term “canned” behaviors. This type of behavior typically represents routine performance of tasks, which do not require anticipation of dynamic situations. Route following is an example of such a task. Route following involves a long-term series of steps and so can model the actions of a pilot or driver in following a route, however routine performance stops short of reacting to dynamic situations, for example recognizing the need for rerouting, generating and negotiating a new route based on weather conditions.

FSMs are the most common way of implementing routine performance. Hierarchical FSMs, wherein each state can contain a lower level state machine, are a common way to implement the dynamic interleaving of standard reactions and routine performance.

2.3.3 **Judgment Calls**

We describe the next two rows in the matrix in reverse order in order to better draw out the distinctions between them.

The most difficult level in the HBR matrix represents judgment calls, problem solving behavior, and non-doctrinal behavior. It is the unavoidable foundation of behavior as it reflects most completely the full range of possible behaviors on the part of human participants in a scenario. The wide range of possible human reactions is typically constrained in software based on judgment calls made by software designers to exclude unlikely outcomes to maintain computational tractability, however in many applications (such as safety studies) it is exactly the outliers in behavior that are of the most interest. Unfortunately, such non-routine behaviors present a combinatorial explosion of possibilities that defeats HBR techniques such as classical control theory, FSMs, decision table lookup, stochastic dynamic programming and LP.

Our experience in the DARPA Command Forces (CFOR), Advanced Synthetic Command Forces (ASCF) and COAA (Course of Action Analysis) programs indicate that it is indeed possible to efficiently model some such behaviors through the technique of hierarchical constraint satisfaction (CSP). The fundamental goal of CSP is to *exploit* the combinatorial explosion at runtime, rather than overcome it at design-time.

2.3.4 Staff Work

In situations where dynamic decision-making is made, there are a large number of more routine follow-up decisions to be made. For example, once the decision has been made *not* to delay traffic but route it one way or another around severe weather, then the particular routes must be computed, waypoints identified, and all the new plans communicated to the effected aircraft or airports, and so on. After the core decisions are made, then the follow-up is comparatively straightforward and procedural, hence the term “staff work.”

This level of processing is typically implemented by basic LP solvers (after the judgmental processes have designed an LP to be solved), simple top-down formatting into multiple copies of interrelated orders, etc.

2.4 *Matrix Application Examples*

2.4.1 NAS Upgrades

We will illustrate the matrix by describing how planned upgrades to the actual NAS fit into it, based on our Task 1 review of the field. In this case we are not representing NAS simulation; rather, we are representing the more active, less routine decision-making on the part of all NAS participants that are being postulated to improve NAS performance by relaxing the operational rigidity of the current system.

The leftmost, heavily crosshatched four regions represent the NAS areas that are getting the most emphasis. The center, lightly crosshatched three regions represent a lesser emphasis. The rightmost, lightest two regions represent the least emphasized.

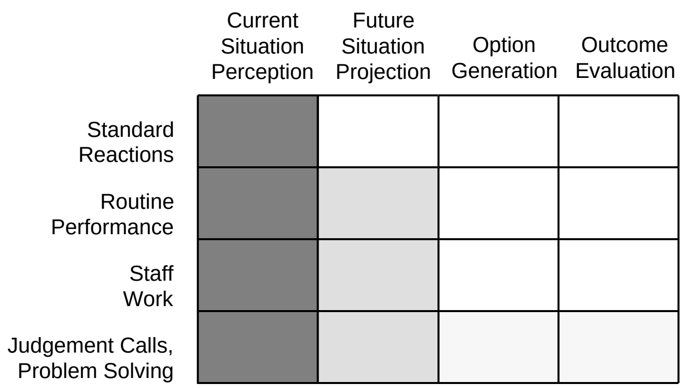

Figure 2.4.1-1 – Focus of NAS Upgrade Plans

The upgrades related to improved navigation aids, Global Position System (GPS) links, precision approaches, etc. fall into the box for pilots' immediate perception of the current situation (i.e., their precise physical location) during routine performance. Similarly, an aircraft monitoring nearby aircraft in order to avoid conflicts falls into the box for current situation perception for standard reactions (though this may also include anticipation of future conflicts, thus pushing it toward the *Judgment Call* category. This change alters the ATM command structure, which used to rely on the ground-based controllers to detect conflicts and inform the aircraft of the problem and what to do about it.

2.4.2 Prior NAS Modeling Emphasis

We further illustrate the matrix by describing how prior NAS human behavior modeling fits into it, based on our Task 1 review of the field.

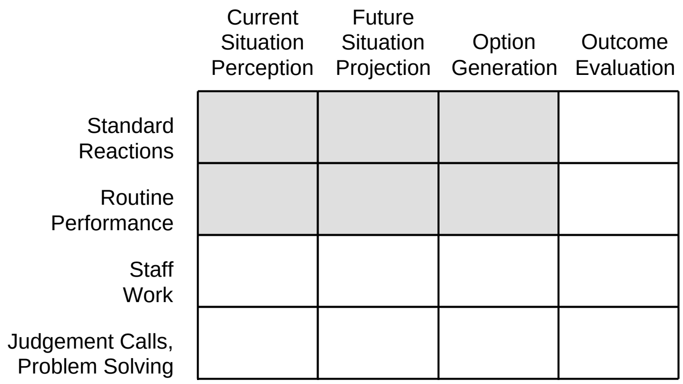

Figure 2.4.2-1 – Prior Focus of NAS HBR Modeling

It is important to note that much of the modeling of humans in prior NAS models was human workload modeling, or human factors modeling. This is of course distinct from the HBR of interest here.

Network based queuing simulation models have dominated HBR modeling for the NAS. This is because even the most casual observer can tell that airplanes queue up for gates and runways, and even the simplest queuing simulations already give valuable insight. This appears to have set the cultural direction and preferred approach. Thus, the basic analysis of the NAS can treat the dynamics of queuing as primary, and modeling judgment of C^3 elements can be handled secondarily.

This places most of the prior NAS work on the second row, with conflict detection and basic route planning extending it to the second and third columns. Simulations that include simulation time conflict detection and two- or three-way maneuvering are placed in the first row.

2.5 Military Modeling Emphasis

Military modeling of HBR has from the very start been forced to deal with judgmental aspects to at least some degree, and this has set the tone and preferred approach for that community. The reason is that maneuvering in the face of the enemy – and hence the high-pressure replanning of maneuvers when the enemy disrupts the plan – has been a dominant military problem for hundreds of years, and has been included in at least some form in every military model. In military operations there is no “steady state,” that is, operations are based on continual disruption and novel circumstances. This differs from, for example, disruptions caused by weather in the NAS. While severe weather systems are large semi-stochastic perturbations to the NAS, they are not intelligently and maliciously attempting to maximally disrupt the NAS, and they are not intelligently bluffing and out-guessing the controllers. An intelligent opponent does all these things, and thus stresses the military C^3 system in ways that make intelligent judgment a primary factor to be modeled. Thus, the blue crosshatched areas in the matrix below represents this aspect of military modeling, in which our team has made significant contributions to the state-of-the-art.

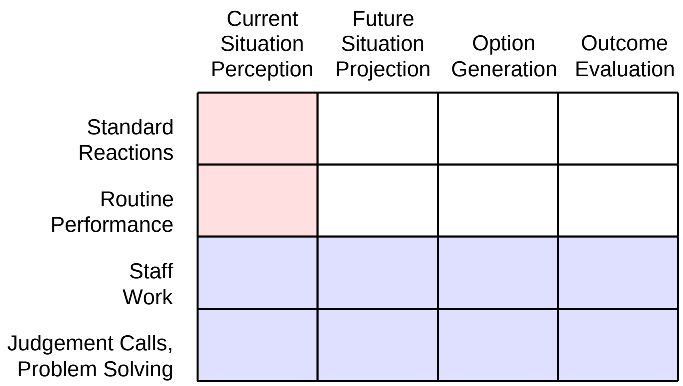

Figure 2.5-1 – Military HBR Emphasis

Combat modeling often uses a simplification of attrition known as The Lanchester/Osipov equations. Even with this simplification, which uses aggregated results rather than playing out of entity-by-entity scenarios, more lines of code are devoted to representing human judgment and planning than to any other aspect of modeling. The moment-by-moment movement, formation keeping, firing at the enemy, avoiding enemy fire, and reaction drills to unexpected attack, and so on are critical to military modeling. However, they tend to all be either fairly short-term actions, or the routine execution of orders from immediately above. Thus, the two red crosshatched areas in the matrix.

2.6 Expanded Matrix

Having defined the basic matrix, we expand its utility by adding two additional dimensions:

- *Multiple layers of resolution*. Each of the regions of the basic matrix can be modeled at varying levels of detail, and so the matrix should take into account the level of detail of the human behavior model. This also includes the level of aggregation of behavior, for example, modeling the behavior of a single controller vs. a TRACON in the aggregate. This latter element includes both more detailed breakdown at a moment in time, as well as resolving differences from one time to another (e.g., dynamically reorganizing sectors and other C^3 relationships).

- *Resolution into subtypes of behavior*. This represents how various elements of behavior interact with each other. This raises issues of how boxes at different levels of complexity and resolution are to communicate with each other.

2.6.1 Multiple Layers of Resolution

A simple representation of multiple layers of resolution is as follows.

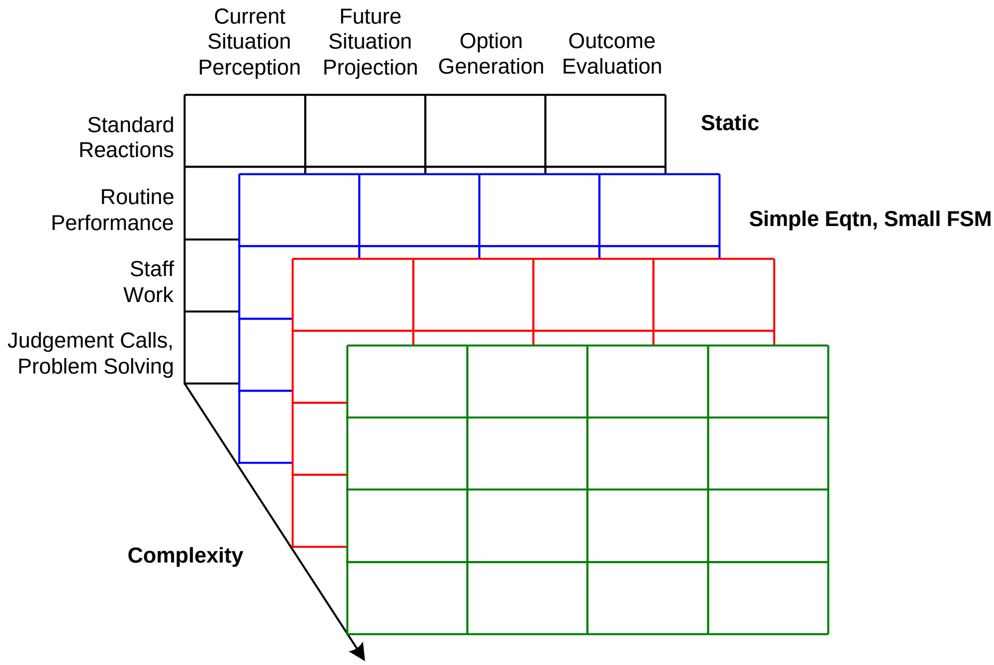

Figure 2.6.1-1 – Multiple Layers of Resolution and Variability

This added dimension deals with the level of detail applied to a particular decision model. Different models may occupy the same space in the two dimensional matrix (e.g., standard reactions/current situation perception) but represent different levels of fidelity and complexity, from static reaction (always take exactly the same reaction to a particular stimulus) to equation-based algorithmic approaches to cognition-based models. This is a fundamental modeling issue, and it appears repeatedly in the field of behavior representation. It is sometimes described as the difference between "real models" and "performance models," between detailed high-resolution models and abstract low-resolution models, and so forth. In behavior models this applies to how much of the internal decision-making processes of the simulated actor are modeled to achieve an overall acceptable transfer function (that is, appropriate response to various stimuli).

As an example, consider a model of pilot behavior during taxiing. Taxiing an aircraft from a gate to a runway can be represented multiple ways. In a high fidelity case, the model may include the following elements:

- A full 4D continuous terrain, so that aircraft can bump up and down on joints of the taxiway, and get off center, etc.
- Modeling the aircraft orientation, velocity, mass, braking power, thrust of each engine, moments of inertia, etc.

- Modeling of the pilot's perception of ground operations and other local aircraft
- Modeling the pilot's operation of the aircraft's controls to achieve desired locations, orientations, velocities, etc.
- Communication of text requests, orders, and answers on different frequencies, including possible miscommunication due to radio interference or misunderstandings.

However, for many applications this level of detail is far in excess of what is required and many of these elements can be eliminated or simplified. For example, the radio model could be replaced with a simpler random communications error model.

At the other extreme of fidelity, control of a taxiing aircraft could be represented as a networked queuing model, where aircraft move from node to node in the network and the runway is a server with fixed cycle time and capacity of one aircraft. The modeling choice, of which overall behaviors are to be broken down into sub-system and controller, or viewed as a single system, is omnipresent.

In addition to the core behavioral representation, there are other elements which push particular models into higher planes of complexity in the matrix. One example is:

Dynamic C3 Relationships: This area models how the patterns of communication in collaborative problem solving change during problem solving. Modification of C3 relationships as events play out is not unusual in combat, as in a hasty fix-and-flank maneuver where two new, temporary C3 elements are spawned. It is to be expected in some future NAS concepts, where dynamic reorganization of sub-sectors is planned. The key technique here is to represent the functioning C3 organization as a data structure, which the model reads (or builds) and executes, not as part of the software architecture of the model. A useful analogy is a program to simulate queuing networks. One would not usually want to hard-code any particular network into the software, but must develop data structures (nodes, edges, networks) that can be built up at run time to represent any particular structure, then executed. This necessitates a more abstract style of programming, where the "model of the network" is not so much a model of a particular network as a toolkit for assembling and running different models.

2.6.2 Decomposition into Behavioral Subtypes

Another axis along which to categorize behavior deals with the nature with which behaviors interact with each other (and in some cases such as agent-based systems, in which sub-behaviors interact to form a single behavior). This distinction is most significant in lowest row of the matrix, judgmental behavior, which in some ways represents the richest repertoire, even though it is the least amenable to closed form numerical analysis. Behavioral sub-types include:

2.6.2.1 Collaborative behavior

This is the behavior of groups, not a single decision maker. In this case, multiple decision-makers are each working on their own individual goals, however they are working in concert towards a larger goal. The set of decisions, and the relationships between them, are generated and asynchronously processed by different decision-makers, but constrained to be consistent with each other at those points where they overlap. An example would be multiple controllers in the same center, working within the bounds of overall flow constraints and respecting each others' capacities.

With one decision maker, the algorithm can pretty much work through the problem in the order it chooses, never worry about deadlock or race conditions, and only backtrack (that is, consider another problem solution) when it "decides" that its own sub problem requires it using in any backtracking approach that looks promising.

All these assumptions fail with multiple decision-makers, necessitating not only a distributed solution procedure but also a public protocol for communicating preferences and partial solutions between agents. The required information is somewhat akin to the numerical LP solution procedure of Danzig-Wolfe decomposition. An instance of such an information exchange among multiple collaborating agents is the Command and Control Simulation Interface Language (CCSIL) initially designed by MITRE Corporation in the STOW project then implemented and extended by SAIC under the DARPA CFOR project (also part of STOW).

One can categorize human behavior models as to how they collaborate.

2.6.2.1.1 Cooperative behavior

The fundamental difference between unitary and collaborative behavior is the need for separate agents to reconcile their different suggestions for how to solve a common sub-problem. For example, in any collaborative process, there must be some way of resolving the conflict when two parties suggest different values for a variable. The way of resolving this problem is different in a purely cooperative system, where both know that both have exactly the same utility function and have every reason to give accurate descriptions of their perceptions and options and goals, and a competitive system, where these assumptions do not hold and different resolution techniques must be used.

In cooperative problem solving, the generic framework for sharing subproblems can be filled in by particular mechanisms that presume a degree of "trust" and mirror-imaging between the agents working on the subproblems. One exteme would be to simply take whichever minimizes the local constraint violations, regardless of which party proposed it. Similarly, two agents laying out a sequence of actions could utilize a "max-max" variant of Von Neumann's "mini-max" algorithm (more likely "max-average-max-average", considering that variables like weather are uncertain, so the outcomes are uncertain).

2.6.2.1.2 *Competitive behavior*

Competitive behavior is the flip side of Cooperative behavior. It is the case where there are multiple independent decision-makers who are working against rather than with each other rather. In this case, each decision-maker must be prepared to plan not against a static or benign situation but rather in an environment designed to thwart achievement of its goals. In military combat this is the classic situation of enemies fighting each other. In the NAS domain this could represent the behavior of multiple airlines competing for resources such as departure slots. Mechanisms such as a synthetic market could be used for coordinating competitive planners in a such a situation. They could also be applied in very short-term competitive problems, as exemplified by the market-clearing mechanisms proposed for computer operating systems.

In military analysis, both real-world and simulation, planning under these conditions is handled by analyzing the various outcomes that could occur for each future action on the part of the decision-maker and consequent possible reactions on the part of the other decision-makers. Having enumerated the tree of likely outcomes, the decision-making agent sets off down a path that promises to offer a good probability of an ultimately successful (from its perspective) outcome. In military terms this type of adversarial planning is called evaluating "branches and sequels," a computational DST along these lines was demonstrated by SAIC under the DARPA COAA program. The extreme version is the original form of Von Neumann's famous "mini-max" algorithm for playing competitive games, which relies on the zero-sum assumption (again, more likely "minimum-average-maximum-average").

One can categorize human behavior models by their ability to handle competitive situations through planning.

2.6.2.2 Non-doctrinal solutions

Non-doctrinal solutions are the "out of the box" solutions that can be generated by creativity without regard to rulebooks or standard procedures. These are the toughest cases to model in a computer under many behavioral modeling paradigms. Most behavior models, be they rules or state machines, can represent arbitrarily complex sets of standard procedures – that is, solutions that have already been thought of – however they will not come up with solutions on their own. Under most paradigms a simulated airplane spotting a runway incursion will only consider aborting its landing if the behavioral developer has explicitly programmed this as one of the plane's options; otherwise the plane may continue to land because it doesn't know what else to do. Certain behavior implementation methodologies can come up with novel solutions, CSP and learning algorithms such as neural nets among them.

One can categorize human behavior models by their "creativity", that is, their ability to generate novel non-doctrinal solutions.

2.7 Architectural Considerations

The matrix, especially with different levels of resolution, presents a daunting set of possible behavioral interactions, using quite dissimilar techniques, both within and between agents. Further, the level of resolution may need to be changed, on a component-by-component basis. This might be necessary to assemble a custom simulation to meet a specific, short-term study problem, where a huge monolithic model would be prohibitively difficult to tailor. It might even be necessary to do so automatically during a run, in order to use sophisticated representations when a sophisticated behavior was required, and conserve resources by using much simpler representations whenever they sufficed.

As is typical with software, making one module (behavior) variable necessitates adaptations in the surrounding software with which it interacts. Thus, having a modular toolkit to represent human behavior is likely to have a spillover effect that pushes the other, purely physical or purely Graphic User Interface (GUI), components toward a more modular and object-oriented design. There are other reasons for having a “toolkit” approach to non-behavior components, as well. A management process such as the High Level Architecture (HLA) and its supporting Real Time Infrastructure (RTI) software is available precisely to manage such real-time interactions between modules. It could be used not only for physical interactions (its original purpose) but also for the shared variables and subproblems of collaborative behaviors.

There is one additional possible axis along which to categorize behaviors, which is not treated in depth here. This is the notion of implementation architecture and deals both with the software structure of a behavior itself as well as its relationship to other models. Within simulation systems behaviors can be the entire model, can be separated out as agents distinct from the rest of the models, which compose an entity (as in the AIRMM architecture) or can be integrated at a peer level with other models.

One aspect to consider is future adaptation. In a simulation with a long anticipated lifespan, it will probably be insufficient to adopt any single level of representation as permanent. For some applications, it might be sufficient to model an airplane on the ground performing a “Get to the gate” behavior as simply progressing through a queuing network to arrive a gate, pausing however long is required at each step. In a different application, where the gate is assigned not well in advance but at the last minute, a very different algorithm (possibly including choosing the gate to use) would be employed. To avoid re-writing the software at each run, it would be necessary to have a uniform interface to the behavior, but differing methods of carrying it out in the simulation. This is pretty close to the definition of an object, and it would be greatly facilitated by the toolkit approach suggested above.

3 Assessment of State-of-the-Art in ATM Human Behavior Modeling

3.1 Existing Models and Projects in Human Behavior

The following are very brief descriptions of the individual simulations or modeling technologies assessed as part of this study. This is by no means a complete list of all human behavior modeling projects, inside and outside of the ATM domain, but these models or modeling architectures were chosen either for how well they represent the ATM domain, or widely recognized state-of-the-art in more human behavioral modeling, or both.

Following the list of descriptions is a chart that displays the assessment of these models on five measures: ease of adaptation, speed of operation, breadth of human behavior, suitability to available computing infrastructure, and ease of extension to probable future needs. (Table 3.1-1)

ACT-R (Adaptive Control of Thought, Rational) – A cognitive architecture, and fully implemented simulation system that models problem solving and learning, and has been applied to complex ATC. Production system architecture with network-like associations among working memory elements. Maintains a declarative/procedural memory distinction, and new production rules are learned by analogy through the process of "chunking." Uses a conflict resolution mechanism based on probability of success, value of the goal, and cost associated with firing the production rule.

Apex – Computer simulation of human cognitive, motor, and perceptual processing. Allows users to create, run, and analyze simulations of human-machine systems. Good for formal task analysis and "what-if" analysis as well as rapid processing of experimental human performance data. Able to represent a broader range of human behavior than Total Airspace and Airport Modeler (TAAM) (See below) and SIMMOD.

CFOR (Command Forces) – A constraint-based real time simulation model of combined-arms operations for Army company teams. Represents the goals and constraints of a command decision process as a set of decision variables to be optimized subject to certain constraints (e.g., determine a route to get from Point A to Point B that uses the least time and fuel subject to constraints in moving through Area Z and minimum speed of Y). Takes into consideration the goals and constraints of subordinate units, communicating with them through a series of defined command and status messages.

COAA (Course of Action Analysis) – Constraint-based tool to support rapid evaluation of alternative courses of action. Uses an approach similar to CFOR to implement a DST for military commanders.

COGNET (Cognitive Network of Tasks**)** – An integrated, cognitive/behavioral modeling method and toolset designed to facilitate the process of applying cognitive models to problems in human user performance/training. Allows for the representation of real-time transactions and multi-tasking demands on attention. COGNET models can be paper-and-pencil analytical models, or a fully executable model through the use of software tools.

EPIC (Executive Process Interactive Control) – A modeling tool that allows for the development and testing of theories of multiple task performance. Designed primarily to develop detailed accounts of human dual-task performance. Has psychologically-plausible perceptual and motor systems that embody much of what is known or hypothesized about these systems. EPIC does not learn.

FACET (Future ATC Concept Evaluation Tool) – An ATM research tool to provide a simulation environment for exploration, development and evaluation of advanced ATM concepts. Addresses airspace modeling. Doesn't simulate behavior of humans, but does work well with Center TRACON Automation System (CTAS) and supports concept exploration.

JPSD – Demonstrated the use of models that dynamically vary their level of resolution during simulation, in order to manage resource demands and support occasional use of high-fidelity models of selected behaviors and dynamics.

MIDAS (Man-machine Integration, Design & Analysis System) – Allows simulation of humans interacting with crew station equipment, vehicle dynamics, and a dynamically generated environment.

NARSIM (NLR ATC Research Simulator) – Simulates aircraft, radar, weather and automated ATC for research and development (R&D) of advanced automated tools and integration of ground and air-based systems.

NASM (National Air Space Modeler) –Nested FSMs to represent flight behavior of individual combat aircraft, as well as their coordinated group behavior.

OMAR (Operator Model Architecture) – Support the development of simulation models of human agents interacting with other human agents, both simulated and real, in executing these complex tasks.

PUMA (Performance and Usability – Modeling technique in ATM) – Predict the impact on workload of changes in working procedures or operational tools.

RAMS (Reorganized ATC Mathematical Simulator) – Simulates various ATC functions, and the entire flight plan in various amounts of detail. Is rule-based, and uses a conflict resolution system. An "ATC event" generator that reports its discrete events or triggers thereby enabling the modeler to program a unique set of activities. Also contains a transparent interface to facilitate statistical studies.

Sensible Agents – A modeling architecture that allows for the development of flexible, responsive, adaptive agents that perceive, process, and respond based on an understanding of both local and system goals. Key concept is dynamic, adaptive autonomy for agents. Allows distributed agents to operate and communicate using an industry standard communication infrastructure (CORBA).

SIMMOD Plus – Performs detailed aviation simulation modeling. Has Network Builder that provides the capability to model multiple airports each having multiple runways, taxiways, gates, deicing areas, staging areas, departure queues and concourses, as well as extremely detailed airspace routes, and sectors. Utilizes the mathematical process of queuing. Good for modeling airport-specific events only.

Soar – A general architecture for building artificially intelligent systems and for modeling human cognitive behavior. Has been used to model many aspects of human behavior such as learning, problem solving, planning, searching, natural language, and Human-Computer Interaction tasks. Soar learns, but does not contained psychologically based theories of perceptual or motor behavior.

Swarm – For multi-agent simulation of complex systems. Basic architecture is the simulation of collections of concurrently interacting agents: with this architecture. Can implement a large variety of agent based models. Allows for the simulation of complex adaptive systems, without being tied to any modeling assumptions.

TAAM Plus (Total Airspace & Airport Modeler)– Simulates traffic for decision support, planning, design and analysis. Great for use as a planning tool or to conduct analysis and feasibility studies of ATM concepts at and around airports. It utilizes the mathematical process of queuing.

Table 3.1-1: Model Assessment

MODEL/ TECHNOLOGY	EASE OF ADAPTATION	SPEED OF OPERATION	BREADTH OF BEHAVIOR REPRESENTED	SUITABILITY TO AVAILABLE COMPUTING INFRASTRUCTURE	EASE OF EXTENSION TO PROBABLE FUTURE NEEDS
ACT-R	Like all cognitive models, the primary difficulty in adapting it is getting domain knowledge into the framework. Has already been applied to widely varying domains.	Depends on hardware available and the complexity of the domain's representation.	Is capable of representing a very broad continuum of human behavior, especially because this architecture learns new rules by analogy.	Written in Lisp, and runs on Windows and Mac machines	Same as "ease of adaptation"
Apex (Refs. 11, 25, 26)	Due to data input, various scenarios can be developed and simulations performed.	Known for its rapid processing analysis & its ability to reduce time/ expertise needed to model. fast, consistent integration of behavior templates.	Simulated Human-in-the-loop engineering design and Intelligent tutoring and decision support systems able to diagnose & anticipate information requirements of human operators.	Practical for widespread use.	Currently, supports external users/developers
ASCF	Like all cognitive models, the primary difficulty in adapting it is getting domain knowledge into the framework. The abstract goals, domain objects, and domain constraints for civilian air traffic control would have to be developed. Organization of collaborating teams would need to be represented. The framework to do these has been demonstrated.	Depends on hardware available and the complexity of the domain's representation. Generated realistic tactical military plans in somewhat faster than real time.	Cooperative multi-agent solution of interrelated problems. Selection of plans to satisfy abstract goals. Generation of ground routes. Estimation of fuel consumption, time of travel etc. on ground routes. Tasking of units based on resource and capability constraints. Plug-in use of subordinate planning / optimization algorithms.	Written in Java for Windows PC. Easily ported to Unix.	Same as "ease of adaptation"
CFOR	Like all cognitive models, the primary difficulty in adapting it is getting domain knowledge into the framework. The goals, domain objects, and domain constraints for civilian air traffic control would have to be developed. The framework to do these has been demonstrated.	Depends on hardware available and the complexity of the domain's representation. Generated realistic tactical military plans in somewhat faster than real time.	Development, evaluation, and selection of coordinated and synchronized plans to satisfy goals. Generation of ground routes. Estimation of fuel consumption, time of travel etc. on ground routes. Tasking of units based on resource and capability constraints.	Written in C++ for Unix and Windows.	Same as "ease of adaptation"

Table 3.1-1: Model Assessment

MODEL/ TECHNOLOGY	EASE OF ADAPTATION	SPEED OF OPERATION	BREADTH OF BEHAVIOR REPRESENTED	SUITABILITY TO AVAILABLE COMPUTING INFRASTRUCTURE	EASE OF EXTENSION TO PROBABLE FUTURE NEEDS
COAA	Like all cognitive models, the primary difficulty in adapting it is getting domain knowledge into the framework.	Depends on hardware available and the complexity of the domain-representation. Critiqued division-level tactical military plans in real time.	The COAA system provides critiques and feedbacks of human-generated plans using a subset of CFOR technology. Estimation of fuel consumption, time of travel etc. on ground routes. Compare resource and capability constraints to tasking of units.	Written in Java for Windows PC. Easily ported to Unix.	Same as "ease of adaptation"
COGNET (Refs. 3,4)	Like all cognitive models, the primary difficulty in adapting it is getting domain knowledge into the framework. Has already been applied to widely varying domains.	Depends on hardware available and the complexity of the domain's representation	Is capable of representing a very broad continuum of human behavior	Runs on Unix and Windows	Same as "ease of adaptation"
EPIC (Ref. 5)	Like all cognitive models, the primary difficulty in adapting it is getting domain knowledge into the framework. Has already been applied to widely varying domains.	Depends on hardware available, the number of concurrent multiple tasks modeled, and the complexity of the domain's representation	EPIC is a human-performance model that accounts for parallel, multiple task performance. There is no learning. These facts make it of limited use in representing a broader continuum of human behavior (e.g. decision making, situational assessment, etc.)	Written in Lisp. Runs on Unix, Windows, and Mac	Same as "ease of adaptation"
FACET (Ref. 7)	Several types of data can be read with FACET for input to change of scenario	Rapid prototyping capability	Models aircraft routing only	Hierarchically compatible with CTAS	Designed with modular software architecture to facilitate rapid integration of research prototyping implementation of new ATM concepts. Software writing in Java and C. It is platform-independent, and can be run on a variety of computers.
JPSD	JPSD is a very large project that has been running for about eight years, and has produced many software packages for multiple purposes. Some are applicable and extendable, but some are not. None were designed for modeling civilian ATC. ITAR restrictions apply.	Real time	The parts of JPSD that deal with simulation of air vehicles handle tactical level behavior of fighter, bomber, reconnaissance, and supporting aircraft. Both individual and collective behaviors. Simulation components are no longer under active development.	The parts of JPSD that deal with simulation used the Modular SAFOR (ModSAF) as its basic simulation engine, on Unix machines. ITAR restrictions are in place on ModSAF.	Not easy

Table 3.1-1: Model Assessment

MODEL/ TECHNOLOGY	EASE OF ADAPTATION	SPEED OF OPERATION	BREADTH OF BEHAVIOR REPRESENTED	SUITABILITY TO AVAILABLE COMPUTING INFRASTRUCTURE	EASE OF EXTENSION TO PROBABLE FUTURE NEEDS
MIDAS (Ref. 12)	Modular w/ the user able to specify which modules are active	Designed to run in a timely manner.	Allows simulation of humans interacting with crew station equipment, vehicle dynamics, and a dynamically generated environment. Emphasis is on operator performance under mission conditions.	Runs on SGI computers.	Has been applied to various scenarios. Written with LISP, C and C++. GUI-based.
NASM	Dependent on technical maturity and organizational availability of Joint SIMulation System (JSIMS)	Runs in real time and fast-as-possible mode.	Tactical level behavior of fighter, bomber, reconnaissance, and supporting aircraft. Both individual and collective behaviors. Not yet finished.	Performs on multiple common platforms, in both stand alone and networked modes.	Same as "ease of adaptation"
OMAR (Ref. 17)	OMAR can operate in a distributed environment, wherein multiple OMAR images, each running on a separate computer, can communicate and interact across a computer network to solve complex, dynamic, computational problems.	OMAR can operate in a distributed environment, wherein multiple OMAR images, each running on a separate computer, can communicate and interact across a computer network to solve complex, dynamic, computational problems.	Models the human operator. Its development focused first on the elaboration of a psychological framework that was to be the basis for the human performance models. Particular attention to the representation of the multi-tasking capabilities of human operators and their role in supporting teamwork activities of operators.	Object-oriented implementation based on Common Lisp. Agent behaviors are represented in Simulation Core (SCORE) language.	Supports research on creation of adaptive interfaces that use intelligent agents to monitor information, alert users of changes/problems, seek out & integrate data from disparate sources, and generate potential solution alternatives. Also used to create intelligent controller nodes that can insert such non-linear effects as human decision-making, precision weapons, and the effects of non-conventional warfare into war games and simulations.
PUMA (Ref. 18)	Each data file is in a human-readable, English language ASCII form and can be edited either within the tool that created it or in the text form within any standard word processor.	Depends on hardware available and the complexity of the domain's representation	Mainly workload assessment - interference and amount of workload (increasing or decreasing)	Family of independent tools with a common 'look and feel' and the ability to exchange data readily.	Mainly to be used for comparative purposes and can be modified for any scenarios, now or future.
RAMS (Ref. 19)	Appropriate for the study of new system concepts. Allows for creation of new rules.	Depends on hardware available and the complexity of the domain's representation	Simulation of gate-to-gate 4D flight trajectory	Written in MODSIM II and runs on Unix plaforms	Same as "ease of adaptation"

Table 3.1-1: Model Assessment

MODEL/ TECHNOLOGY	EASE OF ADAPTATION	SPEED OF OPERATION	BREADTH OF BEHAVIOR REPRESENTED	SUITABILITY TO AVAILABLE COMPUTING INFRASTRUCTURE	EASE OF EXTENSION TO PROBABLE FUTURE NEEDS
Sensible Agents (Ref. 21)	Dynamic adaptation of the autonomy level of a Sensible Agent is performed by the Autonomy Reasoner to both promote efficient problem solving and to resolve conflicts.	Depends on hardware available and the complexity of the domain's representation	Spectrum of autonomy: Command-driven (agent responds to external command), Consensus (agents work as a team to devise actions), Local (agents in charge of planning its own actions), and Master (agent plans for self and others - issues commands).	Supports distributed heterogeneous computing environments and third-party connections.	Supports a multi-platform and multi-language research environment including C++, Java, Lisp and ModSIM.
SIMMOD (Ref. 22)	Adaptable to most situations at airport(s) depending on input.	Dependent on number of aircraft in simulation.	Behavior of aircraft in airspace and/or on ground.	Works with Windows 9x/NT/2000 and provides an open architecture that operates easily with other desktop applications (Excel, Access, Freelance, etc.)	Available as an EXE w/ customizable input & output files.
Soar (Ref. 23)	Like all cognitive models, the primary difficult in adapting it is getting domain knowledge into the framework.	Depends on whether or not learning is activated. Depends on level of detail in cognition model.	Soar is a framework. It can model as much or as little of the domain as desired and affordable.	Versions available for most computing platforms, in C, C++, Lisp.	Same as "ease of adaptation"
SWARM (Ref. 10)	Swarm is a modeling toolkit, not a model. The primary difficult in adapting it is getting domain knowledge into the framework.	Depends on complexity and scale of the model.	Swarm is a modeling toolkit. It can model as much or as little of the domain as desired and affordable. Standard capabilities in swarm are the ability to schedule events for execution, to write code which defines agents' behaviors, and so on.	Various version available for widespread use	Same as "Ease of adaptation"
TAAM (Ref. 24)	Rulebases of most aspects are reconfigurable and can be edited even during simulation runs. Linking w/ other programs is possible via input and output files. Additional packages allow linking w/ other ATM programs such as FAA's Integrated Noise Model.	Strongly dependent on scale (flights/day) and computation time varies approx. w/ the square of the number of aircraft (real & ghost) in the simulator	Behavior of aircraft in airspace and on ground	Runs on Solaris or Intel platforms	Available as an EXE w/ customizable input & output files.

3.2 Assessment and Recommendations

The prior work on HBR in these models breaks down into a few broad categories. The first two categories account for most of the models.

Queuing models and their derivatives: These account for the transition of airplanes from phase to phase along their planned routes, with appropriate processing and capacity delays along the ground taxiways, at gates, and so on. Because civilian aircraft in the NAS tend to fly repetitive routes between cities that do not move, without active attack by intelligent adversaries, it is quite effective to have humans specify all the basic patterns of movement, and leave the "book keeping" of particular movements to the computer. Even in the future, a great quantity of the simulation's processing will be precisely this kind of repetitive, fairly predictable behavior.

Workload models and their derivatives: The estimate the 'workload' on human operators when presented with various levels of stimuli over time, and with different required rates of decision-making. However, they do not model the detailed individual and/or group thought processes of forming a situation assessment selecting, mentally generating multiple alternative courses of action selecting, thinking through the possible outcomes of those courses of action selecting, and selecting a course of action.

Soar is unique in that it is an old, well-established Artificial Intelligent program, applied to many diverse domains, and based on a particular model of human cognition. Unfortunately, it has very high computational demands, so that a fairly powerful machine was needed per individual helicopter when it was used in the STOW project with MITRE. Soar represents the thought process of an individual, and the functioning of a team is emulated by establishing linkage between multiple running instances of Soar (it must be done in the problem-reduction methodology of Soar).

Military planning models: These are focused on the individual and/or group decision-making process. Because military operations are planned over an enormous and dynamic range of environments, the target sets are never the same and often moving during operations, and there is always active intelligent attack, they has always been a need not only for the basic "follow a specified path" behavior, but also for models focused on the cognitive factors of situation assessment, option generation, outcome assessment, and order dissemination.

The queuing and workload models are the dominant category, and they will continue to be indispensable. However, the higher-level cognitive behaviors will need to be added to fully meet the future demands of the future NAS modeling system. The basic technologies can be taken from the military planning models, which have always been oriented toward complicated team behaviors. However, the particular military domain content will need to be replaced with NAS-specific domain content.

4 Roadmap for Integrating Human Behavior Models into an ATM Modeling Environment

4.1 Features/Issues

The goal for integrating human behavior models into an ATM modeling environment is to achieve a flexible, reconfigurable and adaptable to new operational concepts NAS M&S system that is capable of being run large scale, over various levels of resolution, and in a distributed fashion. It is our assessment that no single model or modeling technique will be sufficient to represent the full range of NAS actors and events. Rather, a toolkit should be assembled that will fulfill the requirements of the M&S system. This toolkit should be capable of simulating the full range of gate-to-gate activities involved in complex operational concepts.

The approach should minimize development cost and risk by leveraging significant investments already made by the Government in individual NAS models. The goal is a fully open, interoperable architecture that incorporates specific components of legacy models, or in specific cases, the entire legacy model.

The toolkit should contain proven state-of-the-art agent technology, detailed NAS domain models, a variety of analysis methods, validated mathematical queuing and physics-based trajectory models, and validated human performance multi-tasking models. Even within each of the above-mentioned categories, a variety of approaches should be available, as "agent technology" can be implemented in a number of ways, each having its own set of pros and cons. As an example, "agent technology" may utilize one or more of the approaches listed in Appendix E, and one approach may be more effective than another in a given subset of the ATM domain.

4.2 Timeline

Achieving all of the above requires a layered, time-phased approach.

In the *first layer*, a more intensive study of the models and modeling architectures identified in this study is required, with the purpose of identifying the specific elements that need to be included in the ATM modeling toolbox.

The *second layer* should tackle interoperability issues, including standards such as the HLA. In some cases, code may be essentially copied into a new environment, whereas in other cases, specific communication protocols may need to be adopted and utilized.

The *third layer* should address the interface for an M&S analyst to access the toolkit. Of specific interest here is how the analyst specifies the question of interest, the domain and scenario, and perhaps even the operational concept. The analyst should be able to understand how to create the front end of an analysis, how, and which specific, elements of the toolbox will be used, and how to specify back-end data collection.

Throughout this process, projects relevant to M&S will provide new state-of-the-art capabilities. These projects are again noted as tech feeder programs. Examples of some of these tech feeder programs are listed below:

- Simulation “front end” for experimental/simulation design
- Construction of agent and activity “base pieces” for use in lower and higher fidelity environments
- Translation between levels of resolution
- “Back end” analysis/synthesis of data.

The above programs allow for a sophisticated state-of-the-art environment within which one can accomplish a “cradle-to-grave” exploration of a set of operational concepts or questions.

The following table, Table 4.2-1 is a depiction of one possible timeline for tasks mentioned above, being fed by tech feeder programs during the developmental cycle. The main path of new HBR development focuses first on the development, second on refinement of two kinds of high level group behavior: controllers, then lastly, operational control. Keep in mind that this functional prioritization should be tailored to the needs of the particular operational concepts and technologies being developed by NASA.

This would leverage on-going research on entity-level simulations and behaviors that will be performed by other projects. Similarly, the scenario setup and data analysis tools can be expected to come on line over time, and their requirements could be partially driven by the increasing capability of the HBR. The logic guiding the dependencies depicted below is that the setup tools would be independently developed, while basic HBR and queuing-style simulation was developed, then the automated setup tools adapted and integrated with them. Similarly, with the analysis tools.

The suggested approach is to start with a core functionality of sector controllers, then expand functionality of the sector controllers to include negotiated hand-offs between sectors. The development and integration with the ground/terminal operations controllers would complete the basic structure. The internal logic of the controllers must be developed to cover not only the current top-down command system, but also the “management by exception,” which is expected under future concepts based on free-flight and direct plane-to-plane coordination. We expect the roles of controllers to change, rather than be eliminated. This implies that the hierarchy of controllers will have to be built, as will the HLA-based communication paths between aircraft and controllers, and so on. These implied tasks are omitted here in the interest of space and clarity.

When the controller HBR has been developed to the point of covering the entire NAS in an integrated fashion, the airline operations centers (AOC) could be started. The logic here is that each AOC will cover its own corporate subset of planes, under the overall direction of the controllers. More precisely, each AOC would be using each airline's own objectives and options to control their operations, subject to the controllers' constraints. We anticipate that this could be modeled somewhat like a Stackleberg game from economics, where the controllers are the "leaders" and the AOCs are the "followers."

We expect the controllers to continue evolving and improving even after the AOC's have started. Once the controllers can handle sectors, the AOC's can be trained. Ultimately, it is still logical to improve the controllers' ability to perform resectorization.

.

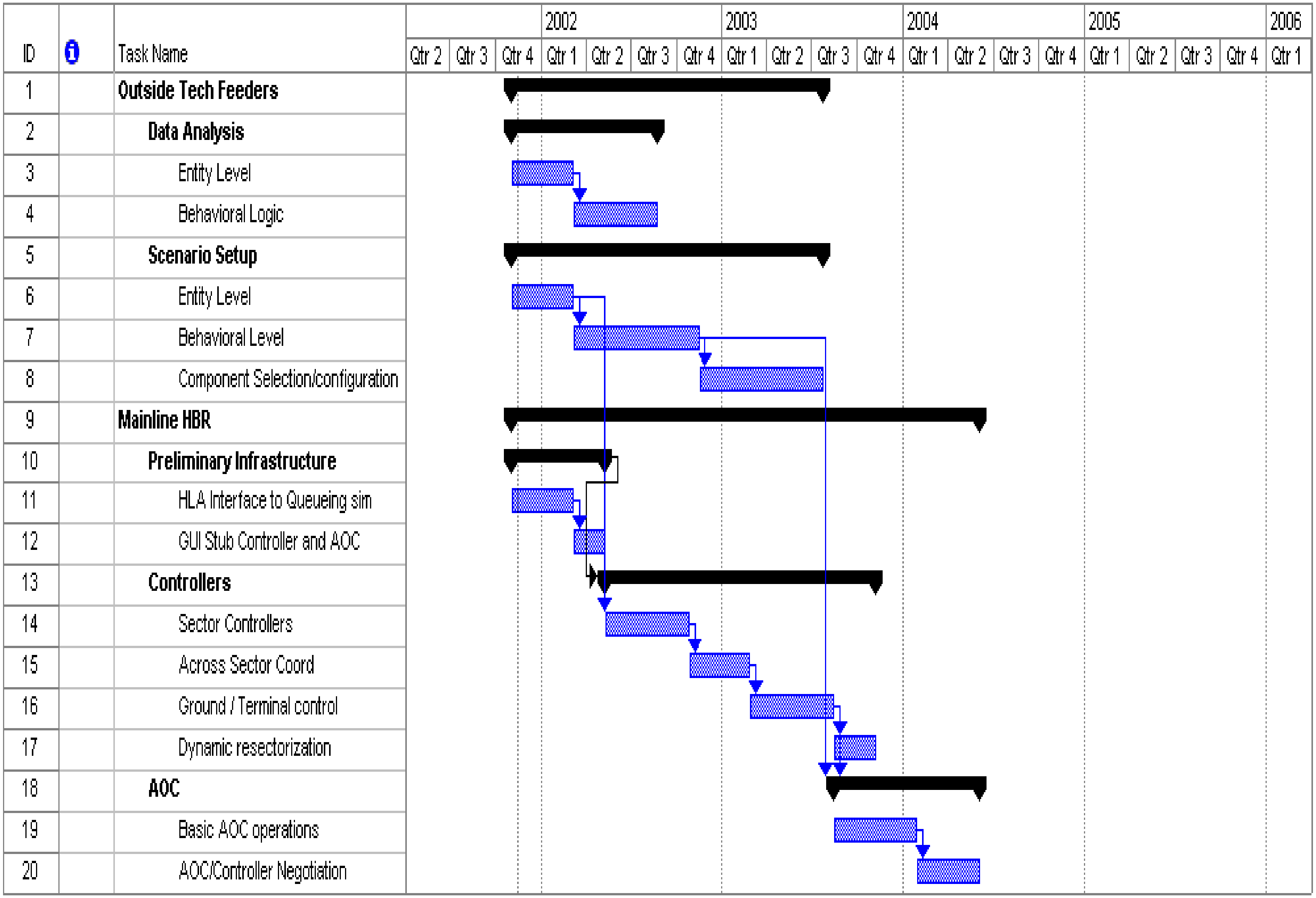

Figure 4.2-1: Phased Approach to NAS Model Development

Appendix A - List of Acronyms

2D	2-dimensional
4D	4 dimensional
AATT	Advanced Air Transportation Technology (Project)
ACT-R	Adaptive Control of Thought - Rational
AOC	Airline Operations Center
ASC	Aviation System Capacity (Program)
ASCF	Advanced Synthetic Command Forces
ATC	Air Traffic Control
ATM	Air Traffic Management
AvSTAR	Aviation System Technology Advanced Research
B&SA	Benefit and Safety Assessment
C3	Command, Control and Communication
CCSIL	Command and Control Simulation Interface Language
CDTI	Cockpit Display of Traffic Information
CFIT	Controlled Flight Into Terrain
CFOR	Command Forces
COAA	Course of Action Analysis
COGNET	Cognition as a Network of Tasks
CRC	Classes, Responsibility, Collaboration (Methodology)
CSP	Constraint Satisfaction Problem
DAG	Distributed Air-Ground
DARPA	Defense Advanced Research Projects Agency
DOD	Department of Defense
DOT	Department of Transportation
DST	Decision Support Tool
EC	Evolutionary Computation
EPIC	Executive Process / Interactive Control
FAA	Federal Aviation Administration
FACET	Future ATC Concept Evaluation Tool
FSM	Finite State Machine
GPS	Global Positioning System
GUI	Graphic User Interface
HBR	Human Behavior Representation
HLA	High Level Architecture
ILP	Integer Linear Programming
JPSD	Joint Precision Strike Demonstration
KE	Knowledge Engineer
LP	Linear Programming
M&S	Modeling and Simulation
MIDAS	Man-Machine Integration, Design, and Analysis System
MIT	Massachusetts Institute of Technology
MRM	Multi-Resolution Model
NARSIM	NLR ATC Research Simulator
NAS	National Airspace System

NASA	National Aeronautics and Space Administration
NASM	National Air and Space Warfare Model
NLR	National Laboratory of Research (Netherlands)
NMCT	Noise Mitigation Controller Tool
NP	Nonlinear Programming
OMAR	Operator Model Architecture
PUMA	Performance and Usability – Modeling technique in ATM
QAT	Quiet Aircraft Technology (Program)
R&D	Research & Development
RAMS	Reorganized Air Traffic Control Mathematical Simulator
RTI	Real Time Infrastructure
RTO	Research Task Order
SAIC	Science Applications International Corporation
SATS	Small Aircraft Transportation System
SCC	System Command Center
STOW	Synthetic Theater of War
SUA	Special Use Airspace
TAAM	Total Airspace & Airport Modeler
TO	Task Order
TRACON	Terminal RADAR Approach Control
UCFIT	Uncontrolled Flight Into Terrain
U.S.	United States

Appendix B – References

1) The Aviation Integrated Reasoning Modeling Matrix, Schwartz, A. and Richards, J. (AIRMM). December 2000.

2) "ATM Modeling and Simulation Architecture Study", Aronson, J. Performed under AATT RTO 70. November 14, 2001.

3) "Cognitive Engineering of a New Telephone Operator Workstation Using COGNET", Joan M. Ryder, Monica Weiland, Michael Szczepkowski, Wayne Zachary, International Journal of Industrial Ergonomics, 1998.

4) COGNET, http://www.chiinc.com/ginahome.shtml.

5) "EPIC: A cognitive architecture for computational modeling of human performance," David Kieras, http://www.eecs.umich.edu/~kieras/epic.html.

6) "Existing and Required Modeling Capabilities for Evaluating ATM Systems and Concepts," Lead author – Amedeo Odoni, International Center for Transportation, MIT, March 1997, http://web.mit.edu/aeroastro/www/labs/AATT/reviews/.

7) "FACET: Future ATM Concepts Evaluation Tool," NASA Ames Research Center, http://www.arc.nasa.gov, Dr. Banavar Sridhar (bsridhar@mail.arc.nasa.gov) & Dr. Karl Bilimoria (kbilimoria@mail.arc.nasa.gov), 2001.

8) "Flight to the Future – Human Factors in Air Traffic Control," Christopher D. Wickens, Anne S. Mavor and James P. McGee, Panel on Human Factors in Air Traffic Control Automation, Committee on Human Factors, Commission on Behavioral and Social Sciences and Education, National Research Council, 1997, http://books.nap.edu/html/flight.

9) Herndon FAA Air Traffic Control Systems Command Center tour information, July 7, 2001.

10) "Integrating Simulation Technologies With Swarm", Marcus Daniels, Swarm Development Group, Santa Fe, NM, October 1999, http://www.swarm.org.

11) "Making Human-Machine System Simulation a Practical Engineering Technique", Roger W. Remington, Michael Shafto, Michael Freed, NASA Ames Research Center, San Jose State University Foundation.

12) MIDAS, NASA Ames Research Center & NASA Langley Research Center, Kevin Corker, Kcorker@mail.sjsu.edu.

13) "Modeling Human and Organizational Behavior," Anne S. Mavor & Richard W. Pew, National Academy Press, Washington, DC 1998, http://bob.nap.edu/html/model/.

14) "Modeling the Capacity and Economic Effects of ATM Technology," Peter Kostiuk, David Lee, Logistics Management Institute, http://atm-seminar-97.eurocontrol.fr/kostiuk.htm.

15) "An Object-Oriented Analysis of Air Traffic Control," Celesta Ball and Rebecca Kim, MITRE Corporation, McLean, VA, August 1991, http://www.caasd.org/library/tech_docs/pre1999/wp90w542.

16) "The Object-Oriented Thought Process – The Authoritative Solution," Chapter 6, Matt Weisfeld, Bill McCarty, SAMS, March 2000.

17) OMAR, http://www.hes.afrl.af.mil/HESS/Programs/OMAR/OMAR.htm.

18) PUMA, http://www.objectplus.co.uk/pumainfo.htm.

19) RAMS, http://www.eurocontrol.fr/projects/rams/docs/Systemoverview.html.

20) "Representing Human Behavior in Military Simulations," Panel on Modeling Human Behavior and Command Decision Making: Representations for Military Simulations, Richard W. Pew and Anne S. Mavor, Committee on Human Factors, Commission on Behavioral and Social Sciences and Education, National Research Council, 1997.

21) Sensible Agents, http://www-lips.ece.utexas.edu.

22) Simmod PLUS! http://www.atac.com/simmod_plus.htm, November 2001.

23) Soar, http://ai.eecs.umich.edu/soar.

24) TAAM Plus, http://www.preston.net/newtaam, November 2001.

25) "Using an APEX Model to Anticipate Human Error: Analysis of a GPS Navigational Aid", Mark Van Selst, Michael Freed, NASA Ames Research Center, 1997.

26) "Using Simulation to Evaluate Designs: The APEX Approach", Michael A. Freed, Michael G. Shafto, Roger W. Remington, NASA Ames Research Center, 1998.

Appendix C – Example of the Human Behavior Matrix in High Fidelity Format

The three following tables illustrate the human behavior matrix presented in Section 2, and illustrates one way it might be filled out for arrival, departure, and en route situation, in a fairly high fidelity format.

Arrival Situations	Current Situation Perception	Future Situation Projection	Option Generation	Outcome Evaluation
<u>Standard Reactions</u> (Emergencies trained for)				
Aircraft/vehicle/animal/object on runway	Pilot sees object on runway, thinks "should I abort?"	Pilot feels aircraft will still be on runway when he should be landing	Go around or no?	Best solution or no?
Engine failure			Go around or no?	Best solution or no?
Gear doesn't come down and/or lock			Go around or no?	Best solution or no?
Approach cut off by another aircraft				
<u>Routine Performance</u>				
Landing procedures			Go around or no?	Best solution or no?
Communications			Go around or no?	Best solution or no?
Taxiing			Go around or no?	Best solution or no?
<u>Staff Work</u>				
Paperwork			Go around or no?	Best solution or no?
Flight tracking			Go around or no?	Best solution or no?
<u>Judgment Calls/Problem Solving</u> (Emergencies NOT trained for)				
Bird strike			Go around or no?	Best solution or no?
Cockpit panel troubleshooting			Go around or no?	Best solution or no?
Electronics failure (Any instruments like navigation, radios, etc.)			Go around or no?	Best solution or no?
Communications failure			Go around or no?	Best solution or no?
Physical/health-related			Go around or no?	Best solution or no?
Fire				
Weather				

Table C-1: Human Behavior Matrix Example – Higher Fidelity

Departure Situations	Current Situation Perception	Future Situation Projection	Option Generation	Outcome Evaluation
Standard Reactions (Emergencies trained for)				
Aircraft landing on same runway as you're cleared for take off	Pilot sees aircraft approaching on runway after being cleared, thinks "should I still go?"	Pilot feels aircraft will be on runway when he should be taking off	Go around or no?	Best solution or no?
Engine failure			Go around or no?	Best solution or no?
Aircraft ahead of you still on runway as you're cleared for take off				
Object on runway (vehicle, animal, aircraft part, etc.)				
Routine Performance				
Taxiing			Go around or no?	Best solution or no?
Take off procedures			Go around or no?	Best solution or no?
Communications			Go around or no?	Best solution or no?
Staff Work				
Paperwork			Go around or no?	Best solution or no?
Flight tracking			Go around or no?	Best solution or no?
Judgment Calls/Problem Solving (Emergencies NOT trained for)				
Bird strike			Go around or no?	Best solution or no?
Problem troubleshooting			Go around or no?	Best solution or no?
Electronics failure			Go around or no?	Best solution or no?
Communications failure			Go around or no?	Best solution or no?
Physical/health-related				
Fire				
Weather				

Table C-1: Human Behavior Matrix Example – Higher Fidelity (Cont'd)

Table C-1: Human Behavior Matrix Example – Higher Fidelity (Cont'd)

En Route Situations	**Current Situation Perception**	**Future Situation Projection**	**Option Generation**	**Outcome Evaluation**
Standard Reactions (Emergencies trained for)				
Routine Performance				
Communications				
Staff Work				
Paperwork				
Flight tracking/navigation				
Judgment Calls/Problem Solving (Emergencies NOT trained for)				
Bird strike				
Cockpit panel problem troubleshooting				
Electronics failure				
Communications failure				
Physical/health-related				
Engine failure				
Fire				
Weather				

Appendix D – Issues in Assessing Human Behavior Models

This document outlines some of the questions and elements to bear in mind when examining human behavior models or technologies or problems, in ATM for NASA's TO 69 under AATT. Although it was beyond the scope of this study to answer in detail all of the questions/issues listed below, it is important to illustrate some of the reasons that the topic of human behavior modeling can be so rich and complex.

Standard Questions in Human Behavior Modeling

This is a list of some of the standard questions that should be addressed in order to gain a thorough understanding of how human behavior is being modeled in a system. This applies to analytical as well as simulation models. Of course, analytical models are limited to simpler situations, as they must be tractable. These issues are important in air traffic control, military airspace management, battle management, economic simulation, and so on. This list is based on SAIC experience in coding models, managing system integration between multiple vendors' products, leadership of national conferences and developers' workshops on HBR, etc.

- The true utility measures of C^3 elements in the system. (e.g. rampers take care of their airline's planes first, so their utility measure is to minimize weighted delay)
- What are the C^3 nodes in the system for each different plausible behavioral scenario to be considered (e.g. pilot, ground controller, en route controller, etc)?
- Range of behaviors required to carry out different plausible C^3 scenarios (and the minimal generalization of those behaviors is what needs to be built into our behavior modeling toolkit)
- What are the meso, macro, and micro levels of behavior? Which are modeled? For those not explicitly modeled, how are the boundary conditions, unmodeled dynamics addressed?
- Time span of perceptions and behaviors (build air ports over years, buy slots over months, assign gates over days, assign taxiways over minutes, adjust ailerons over milliseconds) that need to be modeled
- How do long-term behaviors constrain, limit, or bias shorter term behaviors? When can the long-term behaviors be considered as static "scenario specification" for shorter term?
- Analytic mathematical model versus simulation model. Is the entire model analytical? All simulation models must "bottom out" in analytic approximations (else they would continue down in nearly endless detail until they reached quantum mechanics). Where is that line drawn? How is the uncertainty in the analytical approximation handled (e.g. by random noise on the result)? Could the line be moved up in some modules, in order to simplify the simulation, without obscuring any important interactions? Does

it need to be moved down in some modules? (e.g. many combat simulations would benefit by having their movement and attrition modules greatly simplified, and their commander modules improved).

- Modeling the result of behavior vs. modeling the process of behavior (result: straight and level flight, process: feedback controller attached to dynamic plant model).
- Ground-truth of simulation versus actor's perceptions. How is ground truth communicated between software modules, and how are perceptions derived from ground truth?
- Time stepped vs. tick based vs discrete event simulation
- Are individual behaviors analytically modeled, or rule-based (and all variants thereof), or optimization (and variants thereof, such as genetic algorithms, neural nets, etc.)?
- How are streams of exogenous input streams handled? Examples are weather, passenger demand, etc. what events should be exogenous, and which endogenous?
- Is the system modular enough so that different pieces can be unit-tested? For example, can we run the purely physical simulation before even writing the C^3 parts? Can we test the C^3 parts separate from the physical simulation? Communication separate from decision-making? Formation of situation awareness separate from the sharing of situation awareness?
- Is the system modular enough that different components can have their level of resolution changed? for example, with a given module in en route flight, can we switch (w/o recompiling any code) between point-airports and detailed airports? With a given set of detailed airport models, can we switch back and forth between detailed simulation of en-route flight and simple stochastic delays? Can the level of resolution be dynamically varied while the simulation is actually running?
- Stochastic versus deterministic.
- Are just the physical behaviors stochastic (movement, success in establishing a communication channel), or are also decision-making and communication processes stochastic? For example, if a decision is clearly for X when A << B, and for Y when A >> B, does the system make a random 50/50 decision when A == B, or is there a discrete threshold effect? Does the actual content of communication sometimes get randomly garbled?.
- Is the level of randomness continuously scalable from zero to normal?
- Is the randomness repeatable or unrepeatable?
- How do decision makers handle subordinate or peer entities when they perform normal behavior, undesired but normal behavior (e.g. assess the local situation, decide to discuss orders before acting because they were inappropriate to the situation) , abnormal but physically possible behavior (complete failure to obey orders, acknowledge communication attempts, etc), or physically impossible behavior (vehicles appearing in the middle of the simulation)?
- Is the terrain (e.g. airport structure) easily modified through input files? How about

C^3 **logic** of individual actors? How about C^3 **structure** and the **set** of actors?

- Is the connectivity node-and-link or full 3D? Is it a mixture (e.g. 3D flight, node-and-link taxiways)
- Format of terrain input files
- Format of plan input files
- Coordinate conversion
- Range of behaviors required in a single run:
- Numerical variation (e.g. change movement rate and shooting rate in a piston-style combat model)
- Selection from pre-defined list
- On-the-fly composition from a pre-defined tool kit
- On-the-fly problem solving / planning
- Recording and playback:
- Can the model save state periodically to enable restart with varied parameters? Can the model be paused in the middle of a run, inspected in various ways, and resumed?
- Can the motions of entities be viewed during a run? Saved and viewed after a run?
- Can the reasons for behavior be recorded and presented in human readable form?
- Tactical / operational / strategic levels of C^3
  - Where are the decision makers physically located?
  - What decisions do they make? (e.g. set target flow rates from airports, vs. tell individual aircraft to when to takeoff)
  - How do the higher-level decisions constrain / guide lower level decision makers?
- How do lower level decision makers report deviations (accidental or on purpose) from the suggested/commanded?
- What tools are available to assist in setup of test runs or production runs? (e.g. stochastic terrain generators, synthetic airport generators, traffic demand matrix generators, etc.)
- What tools are available to monitor the model while running (e.g. dynamic 3D image generation, strip charts, etc.) and to analyze the results?

Objects and Behaviors unique to ATM:

The Classes, Responsibilities, and Collaboration (CRC) Methodology focuses on the classes, responsibilities, and collaboration of objects in the domain to be simulated. The following is a rough start at some the objects relevant to ATM that are not already implied by the previous section.

- Vehicle types (speed, size, owner, fuel state, range, wake vortex force and persistence, frequency of different types of equipment problems, distribution of repair times, runway length and strength limits, etc)
- Runways (length orientation, width, strength, surface condition, lighting)
- Taxiways (similar)
- Gates (location, compatible aircraft, owner, scheduled usage)
- Navaids
- Sensing (what terrain / actors / navaids / weather characteristics can be sensed, content, timing, failure probability and modes)
- En route airspace
- En route "highways in the sky"
- Terminal airspace
- Final, takeoff
- Weather (exogenous generation with reasonable statistics, generation with stressful statistics, avoidance, accuracy and type of sensing)
- Communication (channels, sharing, generation, content, interpretation, timing, failure probability and modes)
- Controller/etc heuristics (e.g. minimum 4D separations)
- Controller/etc regions (volumes) of responsibility
- Stereotypical behaviors (e.g. path objects for dogleg, trombone, hold on ramp, etc.)
- Random errors (actual vs. desired motion, location sensing of self and others, decisions, equipment failure, weather, status reporting, communication contents)
- How do disasters happen? For example, Controlled Flight Into Terrain (CFIT), collision, Uncontrolled Flight Into Terrain (UCFIT)? How can we model the factors that make them more or less likely (e.g. controller workload, mean closest-approach distance, mean number of approaches below tolerance per flight)

Appendix E – Discussion of Other General Techniques & Methods

The purpose of this section is to introduce some of the key concepts and technologies used in "state of the art" human behavioral modeling. Some of these concepts and technologies are "newer" than others, but there is a wide body of research resulting from simulation scientists using older approaches in new ways, as well as developing new methods for the modeling of human behavior, and merging various methods together into hybrid approaches. Hybrid approaches are thought to be very useful and powerful, as they have the potential of merging the component methods pros while offsetting each other's cons.

It is beyond the scope of this study to do a full treatment, and compare and contrast analysis with examples, however, we can briefly illustrate some of the pros, cons, and most appropriate uses of some of the methods listed below. The below Table E-1 is not intended to be a complete and exhaustive list of any method appropriate to the modeling of human behavior. Also, we realize, that another simulation scientist may choose to categorize our list in a different manner, as there are some overlaps amongst the various methods listed below.

Our intent in providing this appendix is that a Program Manager, with a basic understanding of a list such as the one provided below, would be adequately prepared to examine and assess simulations, which claim to represent human behavior. It may allow a program manager to more effectively be able to "look under the hood" of such simulations, and examine if the technology being used is suited for the problem or domain it is intended to simulate. It is also important to note that the below techniques may only be applicable to some aspect of human behavior modeling, e.g. for selecting a course of action, or for planning activities, or for addressing the topics of memory and learning, etc.

Table E-1: Technique & Method Comparison

NAME	PROS	CONS	DESCRIPTION
RULE-BASED EXPERT SYSTEM	Good for domains that are relatively simple and well understood and where expertise can be elicited in the form of if-then rules	Is static; doesn't know how to handle situations not explicitly accounted for in the rules; for complicated domains, need a very large amount of rules—whose global effect may or may not be well understood; can be issues in the precedence or sequencing of rules; knowledge engineering in order to construct the rules can be difficult	Consists of a collection of "if-then" rules

FINITE STATE MACHINES	Good for domains which are adequately described as a set of states, and a deterministic transition between states. Fast. Easy for domain experts to understand.	Is static; doesn't know how to handle situations not explicitly represented in the set of possible states; has no concept of memory; has no concept of learning	A technique that models a system as a set of states, and a deterministic transition between states
CONSTRAINT SATISFACTION METHODS	Good for when a choice of discrete variables, subject to well-defined constraints, needs to be made. Easy for domain experts to understand. Easy to model cooperation of teams.	Naïve implementation needs a very efficient algorithm (e.g. intelligent backtracking, or one that takes specific structural features of the problem into account) since constraint satisfaction problems are usually combinatorially large (NP-complete). Therefore, effective implementation requires isolating a few "large" key variables(e.g. an entire "route"), rather than numerous "little" variables(e.g. latitude of the 5th waypoint).	A technique that solves a problem which consists of a set of variables, and constraint relationships between the variables. Modeling cooperation of teams or organizational units.
Q-LEARNING	Does not need a model of the environment. This method guarantees convergence to the "correct answer" if the environment is stationary, the reward is truly a function of actions applied to various states, all state-action pairs are sampled appropriately, and the learning rate is decreased appropriately over time	Must be able to define all states and actions, and be able to measure the reward from transitioning from one state-action pair to the resulting state. No concept of memory.	A type of reinforcement learning; must be able to represent the domain as a collection of states, actions, and payoffs. The goal is for the system to learn the optimal action (rewards in the highest payoff)
ARTIFICIAL NEURAL NETWORKS	Have been used to represent very complicated and nonlinear functions, good at pattern recognition problems. Is a form of distributed, adaptive, nonlinear computing good for "tough" problems defying human reasoning.	No concept of memory. Must be able to represent the domain as a collection of inputs and outputs. Relies on the concept of training an initially random network to correctly map inputs to outputs. Can take a lot of computational power/time to train an ANN. Insufficient data	A type of reinforcement learning. Is a computational construct designed to mimic the way the human brain processes information. Often used in pattern-recognition problems, e.g. the ANN is used to associate a particular set of inputs to a particular output

ARTIFICIAL NEURAL NETWORKS (CONT'D)		may be a problem. May not be able to generalize to "new" cases, depending on the quality of the training data. Sometimes viewed as a "black box" approach, the structure of the network usually doesn't imply any specific information about the nature of the problem.	
EVOLUTIONARY COMPUTATION	Good for complex, highly-dimensional problems with unknown structure. "Good" answers are found relatively quickly if the search space is not too huge	Must be able to accurately measure the "goodness" of any candidate solution to a problem. Can take a lot of computational power/time to find a "good" answer, depending on the complexity (search space) of the problem.	A technique used to seek an "answer" (nearly optimal or optimal solution) to a problem which consists of a finding values for set of variables. Specific types of EC include genetic algorithms and evolutionary programming
FUZZY LOGIC/FUZZY INFERENCE	Good for situations that are not "black and white". The concept of fuzziness allows for smoother transition between categories than for instance, "yes" and "no"	Because of the fuzziness, it can be harder to attribute an outcome to inputs that generated it. Some aspects of the world are "crisp" and not "fuzzy"—a determination is needed. No formal procedure to select actions based on outcome of fuzzy inferencing.	Similar to rule-based expert system, except that the subject of the "if" and the object of the "then" are treated as fuzzy variables
BAYSEIAN NETWORKS	Deals well with lack, or ambiguity of, information; good way to combine historical (empirical) data with expert knowledge	Expert knowledge, in the form of prior probabilities, can be viewed as some as biasing the results too much. Requires a static network of relationships.	A network of variables or "events" that map out cause and effect relationships through the use of conditional probabilities
AGENT-BASED (COMPLEXITY-BASED) MODELING	Good for examining (unexpected) emergent results of a complex adaptive system	Depending on the complexity of the simulation, it may be difficult to attribute cause to effect	A "bottom up" simulation of a domain, with agents interacting with each others and their environment based on simple rules and locally available information. Emergent behavior is the result.

COGNITIVE ARCHITECTURES (e.g. BDI)	Models knowledge representation and agent reasoning in psychologically plausible ways	May not be the most appropriate or efficient way to simulate desired behavior, depending on the specific task or domain being modeled	A simulation architecture that explicitly utilizes a theory of cognition. BDI is a cognitive architecture that can be used to create agents which act on beliefs, desires, and intentions
CONCEPT LEARNING	Good for learning through experience, if the learner is presented with enough positive and negative examples	May be difficult to collect enough positive or negative examples for this technique to be effective	Technique used to learn, through positive and negative examples, a particular concept, including all of the positive examples, and none of the negative examples
CASE-BASED REASONING	Is an example of learning through experience, building theories, and updating them based on specific cases experienced. There is high psychological plausibility to this approach.	May be difficult to construct a broad, well-indexed set of cases for the agent. Success depends on extracting relevant knowledge from the experience.	A problem solving paradigm which attempts to solve a "new" problem, by finding the most similar previous problem solved (case), and reusing the knowledge contained there in the new situation.
MATHEMATICAL PROGRAMMING (LINEAR, INTEGER NONLINEAR, DYNAMIC PROGRAMMING…)	Finding the optimal solution of a linear program is relatively easy, and is guaranteed.	Must be able to express the problem as minimize (or maximize) an objective function, subject to well-defined constraints on the variables. Represents a "neat" and mathematically appealing way to approach a problem but integer and nonlinear programs can be difficult to solve, and may not be particularly well suited for the problem at hand. ILP is NP-complete, and can only handle numerous "small" (i.e. integer) variables.	The operations research method used to determine the minimum or maximum of an objective function, subject to constraints on variables

MARKOV DECISION PROCESS	Appropriate for multi-stage, sequential, decision processes	Exhibits the “memory-less” property, e.g. a transition from one state to the next (given a selected action) is only dependent on the state one is currently in. “Curse of dimensionality” limits it to fairly small problems.	A type of stochastic dynamic programming—is a representation of a reinforcement learning task similar to Q-learning, except that state transition is probabilistic rather than deterministic
GAME THEORY	Effectively represents strategizing and counter-strategizing, in both complete and partial information environments	Naïve versions assume all agents have complete information about each other’s choices, and mutual payoffs for all choices. Sophisticated implementation that represent partial information, or strictly competitive but not strictly zero-sum, games can be slow to solve. Non-strictly competitive games have no known general solution procedure.	A mathematical theory of rational behavior for interactive decision problems
Adjustable Rulesets	Dynamic and adaptive, in that rules can be modified, or new ones created, to adjust to changing circumstances.	The behavior of the rule set can be dependent on choosing the right values for the steering parameters, e.g. it becomes an optimization problem which may be difficult to solve	Similar to rule-based expert system, but the rules are steering parameter-driven, which allows for the modification of rules, or creation of new ones
Probabilistic/Monte Carlo Simulation	Useful when behavior depends on one or more probability distributions. Uses standard statistics to understand the results, e.g. plotted histogram of measures of effectiveness	Sufficiently sampling the space can require intensive computer power	The technique of simulating an event(s) by taking as input randomly chosen values from given probability distributions