Modeling Individual Differences and Stressors
Using the SAMPLE Cognitive Architecture

Pete Aykroyd
John A. Bachman
Karen A. Harper
Charles River Analytics, Inc.
625 Mount Auburn Street
Cambridge, MA 02138
617-491-3474
{paykroyd, jbachman, kharper}@cra.com

Eva Hudlicka
Psychometrix Associates, Inc.
Blacksburg, VA
evahud@earthlink.net

Keywords:
Individual differences, behavior moderators, human performance models, cognitive architectures, modeling and simulation, SAMPLE, MAMID

ABSTRACT: Building on previous work by Hudlicka on the Methodology for Analysis and Modeling of Individual Differences (MAMID), we have developed a generic architecture for modeling the effects of behavior moderators on human decision-making performance. Our approach to Modeling Individual Differences and Stressors (MINDS) employs the generic, parameterized approach of MAMID while applying it to the SAMPLE cognitive architecture. In doing so, we hope to make the behavior moderating capabilities of MAMID accessible to the wider user community of SAMPLE, as well as explore alternative methods of modeling combined moderator effects. Additionally, SAMPLE’s cognitive architecture supports multitasking, which provides another opportunity to apply the effects of behavior moderators to behavior models. In this paper we discuss the similarities and differences between the MAMID and SAMPLE cognitive architectures as well as our approach to modeling combined moderator effects in SAMPLE’s three principal computational technologies: fuzzy logic, belief networks and rule-based expert systems.

1. Introduction
To be useful for testing and training purposes, human behavior models (HBMs) used in modeling and simulation environments need to provide, to the degree possible, realistic and accurate representations of human performance. Recently the behavior representation community has shown increased interest in developing methods of modeling the effects of behavior moderators on human performance (Hudlicka, 2003; Gratch & Marsella, 2002; Hudlicka, 2002; Hudlicka & Pfautz, 2002; Ritter, 2002; Jones, Henninger, & Chow, 2002). Behavior moderators represent a broad range of factors, such as fatigue, heat, exposure to toxins, emotions, personality, or culture, that affect human physical or cognitive performance. HBMs that do not incorporate behavior moderators can provide only a nominal model of task performance and are therefore limited in scope. By including the effects of a range of moderators within behavior models, analysts and model developers can create more realistic, and hence more useful, simulations and scenarios.

Behavior moderators generally encompass two classes of features that can compromise or enhance an individual’s performance: 1) external factors, which include both environmental stressors (heat, noise, vibration) and externally-induced physical or cognitive stressors (physical work, sleep deprivation, cognitive workload); and 2) internal factors, which include cognitive and personality traits (e.g. obsessiveness, skill level, culture) and emotional states (e.g., anxiety, anger) (Pew & Mavor, 1998).

Since behavior moderators do not simply operate at the input/output level of cognitive decision-making (Hudlicka, 2002; Hudlicka and Billingsley, 1999; Ritter, 2002), it is insufficient to simply adjust the output responses of a decision-making model as a function of moderator states. Rather, it is important to recognize the effects that specific behavior moderators (e.g., fatigue versus anxiety) have on individual components of the cognitive process. This realization introduces a requirement for an underlying cognitive modeling architecture within any system that purports to model individual differences and environmental stressors.

For a given cognitive modeling architecture, we must define how specific behavior moderators alter specific structures and processes within the architecture (Hudlicka, 1997, see also Pew et al., 1998). The key
concept here is that we are not trying to moderate behavior or performance directly, on the basis of the moderator factors. This is the approach that has been taken by a number of past research efforts using non-process-oriented task network models (e.g., Fineberg, 1996; Lockett & Archer, 1997). While providing an effective descriptive approach to the problem, it suffers from lack of generalizability across domains, tasks, and factors. But we can recover this generalizability by incorporating a process-oriented cognitive model and by targeting the moderator factors to moderate the intermediate parameters (and structural components if necessary) of the model itself as opposed to direct moderation of the output behaviors.

Building on previous work by Hudlicka on the Methodology for Analysis and Modeling of Individual Differences (MAMID) (Hudlicka, 2003), we have developed a generic architecture for modeling the effects of behavior moderators on human decision-making performance. Our approach to Modeling Individual Differences and Stressors (MINDS) employs the generic, parameterized approach of MAMID while applying it to the Situation Awareness Model for Person-in-the-Loop Evaluation (SAMPLE) cognitive architecture. In doing so, we hope to make the behavior moderating capabilities of MAMID accessible to the wider user community of SAMPLE, as well as explore alternative methods of modeling combined moderator effects. Additionally, SAMPLE’s cognitive architecture supports multitasking, which provides another opportunity to apply the effects of moderators to HBM’s. In this paper we discuss the similarities and differences between the MAMID and SAMPLE cognitive architectures as well as our approach to modeling combined moderator effects in SAMPLE’s three principal AI technologies: fuzzy logic, belief networks and rule-bases.

1.1 MAMID Approach to Modeling Behavior Moderator Effects

The underlying thesis of the MAMID approach is that the combined effects of personality traits and affective states, as well as a variety of cognitive and individual history factors (collectively labeled individual differences), can be modeled by varying the architecture parameters that control both the processing functions and structure of the architecture itself, as well as that of the individual modules that make up a given decision-making model.

Specific personality types (e.g. anxious vs. aggressive) can be represented by specific configurations of the architecture parameters, which then lead to distinct behavioral manifestations under the same external circumstances (e.g., an anxious commander withdraws, aggressive commander attacks, “nominal” commander continues movement, etc.). This is accomplished by varying processing parameters such as attention and working memory capacity and speed, which determine what cues are accessible to further processing; and structural parameters, which control the long term knowledge mediating situation assessment, expectation generation, goal management, and action selection, etc.

The core feature of the MAMID architecture is thus its high degree of parameterization and the ability to encode the influences of a variety of interacting behavior moderators within this parameter space. The MAMID approach was successfully demonstrated in a brigade-level Stability and Support Operations scenario, but the architecture is domain-independent (adapted from Hudlicka, 2003).

1.2 Motivation

In building support for modeling the effects of behavior moderators within a cognitive architecture, the problem is not finding the appropriate effects of moderators on the model - only domain-specific studies can determine this. The problem is, instead, creating a sufficiently flexible framework to support the full range of effects that a modeler may need to represent based on the results of their studies. The success of MAMID’s approach suggests the need to adapt its methodology to a tool-oriented cognitive architecture, such as SAMPLE, for further evaluation.

MAMID is a domain-independent modeling framework that employs Hugin to implement belief net inferencing and a JESS rule-base shell for rule-based inferencing (Hudlicka, 2003). While this approach proved effective, an analyst seeking to explore the effects of behavior moderators would likely benefit from a more encompassing suite of tools to rapidly develop moderated human behavior models.

MAMID’s most salient feature is its generic approach to modeling individual differences via parametric manipulations of the architecture processes and structures. In MINDS, we have adopted and modified MAMID’s generic, parameterized approach to modeling individual differences and have integrated it within Charles River Analytics’ SAMPLE model, a unified cognitive modeling environment supported by a rich set of model development tools.

The SAMPLE cognitive architecture uses the AI technologies of fuzzy logic, belief networks, and rules to create sophisticated models of human performance. In addition, SAMPLE has an accompanying Graphical Agent Development Environment (GRADE), which provides a suite of user-oriented tools supporting the straightforward development, deployment, and analysis of human behavior models. SAMPLE and GRADE’s modular, component-based architecture allow for easy integration of alternative technologies or algorithms. As part of the MINDS development effort, we are enhancing
GRADE to allow the end user to fully specify the effects of individual and combinations of behavior moderators on the underlying cognitive processes within the model, to extract performance-related data from the model at runtime, and finally, visualize that data in a user-driven manner.

2. MAMID and SAMPLE Cognitive Architectures

While the MAMID cognitive architecture resembles that of SAMPLE in a number of its features, there are a number of key differences, discussed below.

2.1 MAMID Cognitive Architecture

The MAMID cognitive architecture is a sequential “see-think-do” architecture, consisting of processing modules that map the incoming stimuli (cues) onto the outgoing behavior (actions).

The MAMID modules consist of the following: sensory pre-processing, translating the incoming raw data into high-level task-relevant perceptual cues; attention, filtering the incoming cues and selecting a subset for further processing; situation assessment, integrating individual cues into an overall situation assessment; expectation generation, projecting the current situation into one or more possible future states; affect appraisal deriving the affective state from the variety of influencing factors: static (traits, individual history) and dynamic (current affective state, current situation, goal, expectation); goal selection, selecting the most relevant goal for achievement; and action selection, selecting the most suitable action for achieving the current goal within the current context. Figure 1 illustrates the MAMID cognitive architecture and mental constructs that comprise input and output of the architecture modules (Hudlicka, 2003).

Figure 1: MAMID Cognitive Architecture and Mental Constructs that Comprise Input / Output of the Architecture Modules

2.2 SAMPLE Cognitive Architecture

SAMPLE is a domain-independent architecture for modeling situation awareness (SA) centered decision-making in high-stress, time-critical environments. While it was originally developed to model the tactical aviation pilot, the architecture of the model itself is domain-independent. It is a general use HBM that has been recently applied to the commercial aviation arena under a number of efforts in the Distributed Air/Ground Traffic Management (DAG-TM) domain (Harper et al., 2002) and to Military Operations on Urban Terrain (MOUT) (Aykroyd, Harper, Middleton & Hennon, 2002).

As shown in Figure 2, information is drawn from a world model, filtered according to the agent’s current attentional focus, and processed via a suite of information processing algorithms, which include sensory processing (e.g., visual, auditory, or haptic modeling) and perceptual processing to generate a set of identified events of interest to the agent. A situation assessment process then translates the low-level events into high-level situation assessments, which are finally processed by a decision-making module to produce actions or communications affecting the world state. Actions chosen by the decision-making module are passed to a goal manager/action selector, which deconflicts the agent’s currently chosen tasks and chooses the set of tasks of highest priority to the agent that can be executed concurrently. Actions taken by the agent that redirect attentional focus are sent back to the attention allocator module, creating a feedback loop.

Figure 2: SAMPLE Cognitive Architecture

Each of the component cognitive modules draws from a suite of internalized mental models of the external world, stored in long-term memory, and used to interpret the world state, identify events and situations, and select appropriate responses. Additionally, each module both draws from and populates a short-term memory representation with identified events, situations, and selected tasks and procedures, which collectively model the individual’s real-time interpretation of the world state. The cognitive processes within SAMPLE are modeled computationally through several AI technologies,
including fuzzy inferencing for information processing, Bayesian reasoning for situation assessment, and rule-bases for decision-making.

2.3 Differences in Cognitive Architecture
As an early version of MAMID’s cognitive architecture was based on an early version of SAMPLE, there are close resemblances between the two: both are recognition-primed decision-making models (Klein, 1989) incorporating the sequential steps of information processing, situation assessment, and decision-making. However, there are a number of key differences.

The most critical difference is MAMID’s explicit modeling of the cognitive appraisal of affect, which enables it to dynamically generate four emotions, and in turn model their effects on cognitive processing. In SAMPLE an affect model is not explicitly included but its function can be performed by a belief network modeling basic or complex emotions.

MAMID places goal management before decision-making: during the goal management step, the agent chooses the most relevant goal for achievement, and subsequently selects the most suitable action for achieving the current goal within the current context. This architecture assumes that the agent will only perform a single task at any given time. The SAMPLE model enables multitasking by placing goal management after action selection. This allows the agent to generate a set of desired actions that may or may not be able to be executed simultaneously. The set of actions are passed to the goal manager/action selector, which produces a deconflicted set of actions that includes parallel tasking targeted to achieving the highest priority goals that can be pursued simultaneously. In both MAMID and SAMPLE, while the developer defines the goal hierarchy, the agent can change relative priorities of goals at runtime based on the current situation or moderator state.

Finally, MAMID incorporates an explicit expectation generation module to project potential future outcomes of the current situational context. In SAMPLE this function is considered part of situation awareness, specifically level 3 SA (Endsley, 1995), and is implemented within that module.

3. Implementing Behavior Moderators in SAMPLE
MAMID laid the groundwork for moderating agents by modeling the effects of individual differences (cognitive and personality traits) and emotions on an HBM (Hudlicka, 2003). However, MAMID, at this point, lacks a mature set of integrated tools to easily allow a user to construct, verify, and visualize agent performance. By extending the SAMPLE cognitive architecture, we sought to provide the capability to explore behavior moderators within an existing set of agent development tools. This extension required that the SAMPLE cognitive architecture be parameterized sufficiently to represent both the effects of a single moderator at varying levels of intensity and the interactions between moderators.

Realizing that we could not anticipate the appropriate means of modeling the effects of all potential moderators, and combinations thereof, on a given HBM’s behavior while maintaining a generic architecture, we sought to incorporate sufficient flexibility in the available parameters to allow a modeler to use research results to drive the design of moderator effects on a case-by-case basis. To support this flexibility, we did not differentiate between traits (e.g. extraversion, neuroticism) and states (e.g. fatigue, anxiety) in their effects as MAMID did, instead leaving that decision to the model developer.

The model developer specifies moderators as well as minimum, maximum, and default values for each. A nominal moderator is also defined by default. However, moderator levels need not represent fractions of the model’s aggregate moderator-state. For one model, it might make sense to create a moderator for which zero was the default level and 1, the maximally moderated level. However, it is just as easy to envision a different moderator (e.g. introversion/extraversion) with a scale that is extreme at the minimum and maximum values (e.g. 0 = introverted; 1 = extroverted; and 0.5 = nominal). Currently, we provide parameters to adjust the model’s cognitive processing capacity as well as its long-term memory structures based on these moderator values.

3.1 Moderating Cognitive Processing Capacity
Cognitive processing speed and capacity refer to how quickly mental operations, such as the processing of cues from the environment, can be performed, and the amount of information that can be processed simultaneously by the model. Differences in processing capacity can be caused by traits (e.g. general intelligence) or due to dynamic factors (e.g. anxiety or fatigue). For example, a moderator could reduce the number of external cues that an HBM can detect and process in a specified time interval, or the number of situational factors that can be considered in making a decision. MAMID provides this capability by including parameters to modify the model’s working memory and processing speed. We have incorporated a similar mechanism for modeling processing speed, but have not yet implemented an explicit model of working memory that can be moderated. SAMPLE moderates capacity by providing interfaces to modify: 1) the number of cues that can be received in a given period of simulation time by a specified cognitive process; and 2) the capacity of specific cognitive processes based on behavior moderator state.
The first capability allows the modeler to limit the number of messages that can be received by a specific component of the model (e.g., a belief network performing a given SA function) in a given time interval. These input rates can be set for individual behavior moderators, thus allowing for detailed fine-tuning of model responses. When more than one moderator is active, their effects should be combined. Currently, the effect of a combination of moderators is modeled by selecting the slowest input-rate of the active moderators. However, this implementation does not yet provide sufficient modeling flexibility. Since it is unclear how best to combine these rates, there should be sufficient flexibility to support differing models of this interaction. In future development we will address this by providing the modeler with a range of strategies for combining input rates (e.g., MIN, MAX, average, or weighted average).

The second effect, limiting processing capacity of individual cognitive processes, is implemented using mechanisms specific to the AI technology being used (e.g., belief network, rule-based expert systems). A separate interface is provided for each technology component allowing the modeler to configure the effects of specific moderators on the process. For example, the rule-base could enable the user to limit the number of rules that can fire in a specified time interval. Here SAMPLE differs significantly from MAMID, which achieves this effect by manipulating mental constructs (e.g. cues, situations, goals, etc.), whereas our moderation of cognitive capacity is tied to the specific representational technology.

3.2 Moderating Long Term Memory

Much like MAMID, we have modified SAMPLE to represent the effects of moderators through structural modifications to long-term memory. MAMID accomplished this by providing alternate belief nets and rule clusters for particular combinations of static behavior moderators (e.g., skill level, personality trait, individual history). In SAMPLE we have attempted to provide a mechanism to incorporate these effects without requiring separate model instances (e.g., belief nets and rule clusters). Furthermore, because we do not explicitly differentiate between states and traits, long-term memory modification can be used to model the effects of both dynamic states and static individual differences.

In SAMPLE, we support long-term memory modification of belief networks and rule-based expert systems, as well as for our fuzzy logic engine. Our mechanisms for modeling moderators also enable the parameterization of interactions between moderators, and can model the effects of moderators at varying levels. The following three sections discuss the extensions of our belief network, rule-base, and fuzzy logic engines to support moderation.

3.2.1 Belief Network Moderation

MAMID provided the capability to modify long-term memory structures modeled by belief nets by defining separate BN structures for each moderator or combination of moderators. SAMPLE does not require separate belief networks for each moderator or combination thereof. Instead it represents each moderator as a separate node in the network, which can then be used to appropriately alter the behavior of the network. Like other BN nodes, each moderator node has a number of states (e.g., high, medium, low), which are defined by the modeler based on the specifics of the particular application. This provides the flexibility to use as many states as can be supported by research. For example, if evidence existed to support modifying the model’s behavior based on three levels of fatigue (e.g., very fatigued, moderately fatigued, and unfatigued), then the modeler would define those states in the fatigue moderator node. However, if evidence only supported modeling of fatigued and unfatigued behavior, then only those states would be added to the moderator node. Once the states are defined, fuzzy membership functions are created to map the moderator’s raw value into a belief vector.

Since a node represents each moderator, the modeler can use the evidence from that node to appropriately alter the beliefs of other nodes. Figure 3 shows an example belief network used to model a commander’s assessment of his unit’s combat effectiveness. Without any of the moderators active, this simple BN aggregates the beliefs of how high or low the unit’s attrition state is into a personnel node, which represents the rating of the unit’s manpower; good or bad. This is combined with the belief of the adequacy of the unit’s number of operable vehicles to create the commander’s belief of the unit’s combat effectiveness. To this BN is added three moderator nodes: LowE-HighN, HighE-LowN, and Anxiety. The first two are static moderators, representing the personality types Low Extraversion – High Neuroticism and High Extraversion – Low Neuroticism respectively. Anxiety is a dynamic moderator with a level that adjusts based on circumstances. These nodes can then be linked into other nodes in the BN to modify the model of the commander’s assessment.

For example, a Low E-High N personality type is biased towards producing more pessimistic situation assessments (i.e. the loss of a vehicle is regarded as being more detrimental to combat effectiveness than otherwise) (Harper et al., 2005). The opposite is true of High E-Low N personalities. This judgment bias can be modeled by linking the moderator nodes to the combat_effectiveness node and modifying the conditional probability table (CPT) appropriately. Furthermore, with anxiety linked to the combat_effectiveness node, the interactions between these personality types and varying levels of anxiety can...
also be defined. Moderators are treated like other evidence in the belief net.

Figure 3: Moderated Belief Network

This approach enables three important capabilities for modifying the belief network model:

- The value of a node can be biased toward a desired state based on the value of a moderator node.
- A node’s effects on the network can be eliminated by ignoring its state. For example, a belief network could be designed such that when anxiety is high, the value of the personnel node will not influence the evaluation of the combat_effectiveness node. This effectively removes the node from the network.
- The resolution of a node can be reduced by effectively eliminating one or more of its states. For example, the CPT table for combat_effectiveness could be structured such that when anxiety is high, the low and acceptable states of the operable_vehicles node both cause the same negative appraisal of combat effectiveness. This could be used to model situations in which a moderator leads to a less nuanced appraisal of the situation.

Granted, adding moderator nodes to even a fairly simple belief network such as the one shown in Figure 4 can result in complex CPT tables that are difficult to configure. However, since this approach allows us to effectively alter the structure of the BN based on moderator node states and to specify the effects of a combination of moderators, we believe that moderator nodes are a compelling way of modeling the effects of a broad range of moderators on long-term memory represented by belief nets. That said, in order for this mechanism to be effectively leveraged, it must be usable even in complex BNs. To address this, future work will focus on building tools to automate parts of the process of creating the CPT and otherwise manage the complexity of large CPTs.

3.2.2 Rule-Base Moderation

Our design of the moderated rule-base shares goals similar to those of the moderated belief network engine. MAMID’s moderated rule-base relies on entirely separate sets of rules to represent the distinct long-term memory structures associated with particular combinations of individual differences and dynamic states. This approach, however, requires a separate cluster of rules for each combination.

SAMPLE enables different combinations of moderators to be modeled by a single rule-base by allowing the modeler to assign a strength to each moderator-rule pairing. As shown in Figure 4, strengths can be used to ensure that certain rules will never fire if the agent is moderated fully by a specific effect. For example, the second rule has a strength of zero for both anxiety and LowE_HighN, ensuring that it will never fire in an agent moderated by those two effects.

Figure 4: Rule-base Strengths

Using this approach also allows the modeler to create rules that will fire only when moderators are combined. Table 1 illustrates how strengths could be specified in such a way:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Nominal</th>
<th>Anxious</th>
<th>Fatigued</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal Rule</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Anxious Rule</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Fatigued Rule</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Anxious + Fatigued Rule</td>
<td>0</td>
<td>.75</td>
<td>.75</td>
</tr>
</tbody>
</table>

Table 1: Example Rule Strengths

These rules are set up such that when the agent is either in a nominal state or moderated completely by anxiety or fatigue, a rule for that state will fire, mimicking the capabilities offered by defining separate rule-bases for each moderator. However, when fatigue and anxiety are combined in sufficient quantities, the Anxious + Fatigued Rule will fire. Of course the notion that levels of anxiety and fatigue can be quantified with percentages may be artificial. Under our current implementation, the burden
of placing these percentages into semantically meaningful categories that could be supported by empirical studies is left to the modeler and no categorization is explicitly supported. However, it is essential to be able to represent different behavior based on varying levels of each moderator and combinations of them (Pew et al., 1998). Therefore, the rule-strength method’s flexibility makes it a compelling implementation. In further work, we plan to provide high-level tools to users allowing them to take advantage of the flexibility offered, while minimizing the complexity of configuration as well as investigating the applicability of providing explicit categorization of the moderator levels. Furthermore, we will examine whether this method provides sufficient support for modeling effects on behavior based on varying moderator intensities.

3.2.3 Fuzzy Logic Moderation

To create a fuzzy logic engine that is fully modifiable based on levels and combinations of moderators, we believe the following two requirements must be met:

- **Modified fuzzification:** the ability to modify, add, or remove membership functions for both input and output variables in the fuzzy inferencing engine.
- **Modified rule selection:** as in the moderated rule-base, the modeler should be able to specify different fuzzy rule strengths for different moderators or combinations of them.

We addressed the first requirement by allowing the user to specify different membership functions for each moderator. Then, based on the percentages of the active moderators, a weighted average is produced. For example, if a speed of 65 MPH produced fuzzy membership values of 0.7 fast and 0.3 slow nominally, and 1.0 fast, 0 slow for the anxious moderator, then an agent that was 50% nominal, 50% anxious would produce a membership value of .85 fast, .15 slow. A valid concern raised by this method of combining moderators is whether this sort of averaging is meaningful. More research needs to be done to fully understand how moderators are combined in the process of translating raw signals from the environment into semantically meaningful symbolic values. We believe that our approach represents a reasonable starting point to encourage further exploration.

To address the second requirement, we use rule strengths in the same way as we did with the rule-base. This approach has the same strengths (i.e. flexibility) and weaknesses (i.e. complexity to define) as in the rule-base. We believe the correct way to approach this problem is by providing tools to help manage the complexity rather than selecting a different means of implementing the moderator effects.

4. An Example Moderated Agent

To demonstrate our approach to modeling the effects of behavior moderators on human decision-making strategy and performance, we will describe an agent developed using moderators and each of the three core components previously discussed. This example is based on an agent developed using our framework. However, it was created before the rule-base component had been moderated. Since the purpose of this paper is to discuss the potential of our approach for developing models of moderated behavior, not how that agent was developed, we have included an example of how we could have used a moderated rule-base. Section 4.1 describes the nominal agent. Then, in section 4.2, we discuss how specific behavior moderators (based on extraversion, conscientiousness, and neuroticism) were added.

4.1 Nominal Agent Development

This agent was developed to study the effects of moderators on the decision-making behavior of tank platoon leaders conducting a movement in a potentially hostile SASO environment. Our characterization of how various behavior moderators affect behavior are derived from a literature search summarized in the final report for the project (Harper, 2005). During the scenario, the agent is faced with situations including broken-down vehicles, blocked roads, and ambushes that force it to consider the unit’s combat effectiveness, the threat to his unit (in the case of an attack) or to being able to arrive at the objective on time (in the case of a damaged vehicle or road block) and make a decision that handles the situation in a way that maximizes the unit’s ability to complete its mission.

The agent is divided into five functional sections: Event Detection, Situation Assessment, Expectation Generation, Decision-making, and Goal Management (see Figure 5). We describe each function in detail in the following subsections.

4.1.1 Event Detection

The event detection part of the agent consists of a single component, the Event Parser, which was simply an expedient for developing the agent. The Event Parser...
implies a set of simple rules that classify the particular event in terms of the time associated with stopping to address a non-combat problem (e.g., repair a damaged vehicle), the time to call for support to address the problem, and the potential effect on combat strength should the problem not be addressed. It also receives information in the event of an ambush such as size of the enemy element and the method of attack, which it passes on to the appropriate components within the Situation Assessment module of the agent. The data provided by the Event Parser to the rest of the agent forms the basis for assessing the situation and consequently selecting a course of action.

4.1.2 Situation Assessment
This module maintains several situational variables describing the overall operational context using a number of computational components:

- The Mobility component generates an assessment of the convoy’s mobility state as a function of weather and specific terrain elements.
- The Mission Status component assesses whether the platoon is on time, late, or is ahead of schedule based on current mobility assessment, range to objective, and time left before the deadline.
- The Relative Time component assesses the effect of a potential course of action on the mission timeline (e.g., taking 24 hours to build a bridge will make the platoon miss its mission deadline).
- The Combat Effectiveness component tracks the convoy’s current level of combat effectiveness as a function of number of casualties and disabled vehicles.
- Finally, the Threat Condition component assesses the current threat to the platoon based on enemy and civilian presence, the likelihood of mines, and how much enemy contact has already been made.

4.1.3 Expectation Generation
This module maps very to the Level 3 SA discussed earlier. Based on output from the event detection and situation assessment components, the C2 agent then predicts the effect that a number of possible courses of action (e.g., repair, call for support, continue without addressing event) might have on the convoy’s future status and ability to achieve its objective. This is captured within the Cost Analysis component, which assigns a relative expected cost to each of the primary choices.

4.1.4 Decision-Making
The C2 agent’s course of action selection process is managed by three separate components, which map to the three goals of the agent: Engage Threat (or survive), Achieve Objective, and Maintain Convoy. Each rule base makes decisions that support their particular goal. For example, in the event of an ambush the Engage Threat component selects a course of action such as assault or break contact to deal with the threat. In the same circumstances, if the platoon was behind schedule, the Achieve Objective component might select the fight through course of action, since this plan would take the least amount of time and keep the platoon moving forward. The Maintain Convoy component selects courses of action relating to disabled vehicles and road blockages, determining whether the problem should be handled by the platoon or by calling a support unit forward, in essence sacrificing some time for greater chances of success in fixing a problem.

4.1.5 Goal Management
Because the commander must weigh the potential costs and benefits of particular courses of action in terms of maintaining the combat effectiveness of the convoy and achieving the mission objectives, the Goal Manager component within the C2 agent provides the final arbitration among potential courses of action. If multiple decision-making components within the agent “fire” contradictory course of action requests (e.g., stop and repair damaged vehicle versus continue mission), then the goal manager will select the appropriate course of action based on the agent’s current highest priority goals (which can be situationally derived).

4.2 Moderated Agent Development
Now that we have provided an overview of how the nominal agent is structured, we will discuss how moderators can be introduced to this agent. Since the exact required data regarding these effects within an operational domain are typically not available, we combined existing empirical evidence with the judgments of subject matter experts and extrapolated as necessary to make meaningful model modifications. For example, since there is no data on the effects of extraversion on threat assessment in SASO environments, we assumed that since high levels of extraversion correlate with impulsiveness (although correspondingly low levels of neuroticism are generally also required) it may be reasonable to assume that higher-extraversion individuals will tend to be more impulsive and hence more likely to assess a particular entity as hostile. Our model introduced three moderators to the agent: Low Extraversion/High Neuroticism (Low E-High N), High Extraversion/Low Neuroticism (High E-Low N), and conscientiousness. The following sections describe how the model is augmented to represent these moderators.
4.2.1 Modeling Low Extraversion/High Neuroticism

The Low E/High N moderator, in general, has the effect of producing more pessimistic assessments from the agent. Risks are viewed as being of greater in potential severity as well as likelihood. One way to represent this sort of behavior would be by moderating the rules to make a Low E-High N moderated agent always break contact during an attack or always call for support to fix a problem. However, it is almost certainly an over-simplification to say a commander with those personality traits would always do one thing. Instead, by moderating the belief networks that assess combat effectiveness, threat, cost analysis of fixing a problem, ignoring it, or calling for support, we can derive more dynamic behavior. For example, in the Combat Effectiveness BN, this moderator causes the agent to have a lower threshold for KIAs and WIs before assessing personnel strength to be poor. Furthermore, it requires higher numbers of operable vehicles and available personnel to evaluate combat effectiveness as good. We model these tendencies by using the ‘LowE_HighN’ moderator node’s state to bias the CPT tables according to those trends. This causes smaller losses of personnel or vehicles to lead to worse evaluations of combat effectiveness, which in turn is used by the Engage Threat rulebase to select a course of action to respond to enemy contact. This combined with similarly cautious judgements of threat leads to an agent that is more likely to select a conservative course of action in response to the enemy. Similar modifications to the Cost Analysis BN bias the agent toward choosing to handle all but the most trivial problems with support units.

4.2.2 Modeling High Extraversion/Low Neuroticism

The High E/Low N moderator has effects on the agent that are the converse of those of the Low E/High N moderator. This moderator produces more optimistic situation assessments. Risks are minimized, while potential for reward is amplified. This moderator can be represented with exactly the same techniques used to create the previous moderator. However, since thus far the example has only used moderated BNs, we will instead discuss how the rulebase is modified to model a moderator’s effects. As was discussed in the previous section, it is generally not advisable to model a moderator by simply altering the decision-making component of an agent. More nuanced decision-making models can be achieved by instead altering the assessments that feed the decision-making component. However, in this case, we introduce a new behavior to the agent. Nominally, the Engage Threat component can select from a few courses of action: assault, defensive engagement, and break contact. Fighting through is an option that is only output by the Achieve Objective component in response to a tight timeline. However, a reward-seeking, risk-tolerant commander might well react to contact by fighting through, accepting the heightened risk of not directly attacking the enemy in exchange for a better chance of arriving at the objective on time. To accomplish, this we can add a rule to the Engage Threat rulebase that will select that course of action in situations where the threat is relatively low. Then in the rule strength table discussed in section 3.2.2, we set the strength of that rule to 1 for the High E-Low N moderator and 0 for the other moderators. Now in contact where the threat is assessed as low, a High E-Low N will choose to fight through as opposed to assault as agents moderated by other effects would.

4.2.3 Modeling Conscientiousness

In addition to modifying belief network nodes and rulebases, we can also moderate the data fuzzification process. For example, Figure 6 below illustrates the modifications made to the interpretation of enemy presence in order to model conscientious commanders. In this particular example, we see the fuzzy interpretation for the nominal agent on the left, where the enemy presence is categorized as none, individual, or platoon plus. Since conscientiousness induces more precise representations of particular data components (see Figure 6), we increase the number of categories for the conscientious agent, as shown on the right. This agent now categorizes enemy presence as none, individual, squad, platoon, or company. Using this higher level of resolution in the classification of the enemy force, the conscientious agent can differentiate between enemy presence that is similar in size to its own unit (e.g. platoon) and enemy presence that greatly outnumbers its own unit (e.g. company), thus allowing the agent to be more strategic about attacking larger units while remaining willing to attack equal sized units.

Figure 6: Enhanced Enemy Presence Interpretation for Conscientious Commander

5 Conclusion

The ability to model the effects of individual differences and stressors is of key importance in producing high-fidelity human behavior models. MAMID’s initial work in representing these effects not only demonstrated the benefit of moderated models, but also provided an
important framework on which to build. In MINDS, we have incorporated the MAMID concepts of parameterization and structural modification of long-term memory, but have extended this approach to our fuzzy logic engine and have explored a different approach with belief nets and rules. Are approach links the moderator effects more closely to the actual implementation than does MAMID, which focuses on higher level psychological constructs. Furthermore, we are focused on developing tools that operate in GRADE to allow modelers to configure moderated components to represent a broad-range of effects. The greater the flexibility of the model, the more useful it will be to future cognitive modeling efforts.

6 Future Work

Our continuing work on the MINDS project has two principal goals. First, we are designing and implementing an attention allocation mechanism that can be effectively parameterized based on the effects of behavior moderators. We will also focus on providing the user with tools to manage the complexity that arises in defining behavior moderator effects. We are still not confident that our approach to moderating the fuzzy logic engine is flexible enough to support the full range of effects a model developer may require. We are continuing to explore this issue.

7 Acknowledgements

This work was performed under US Army contract number DAAD16-03-P-0056 with the Natick Soldier Systems Center as well as USAF contract number F33615-03-C-6342 with the Air Force Research Laboratory, Human Effectiveness Directorate (AFRL/HE) and the Defense Modeling and Simulation Office (DMSO). The authors thank Mr. Robert Auer and Mr. Dave Tucker at the Natick Soldier Center and Lt. Matthew T. Eaton and Dr. Mike Young of AFRL/HE for their support and technical guidance.

8 References


Author Biographies

PETE AYKROYD is Software Engineer at Charles River Analytics. Mr. Aykroyd’s areas of interest include object oriented software design, adaptive visualization, and agent-based systems. He is currently the lead software engineer on the Modeling Individual Differences and Stressors (MINDS) project, where he is developing a framework for creating moderated behavior in human behavior models. He received a B.A. in Computer Science and Classical Civilizations from Colby College in 2000, after completing a senior thesis project studying computer generation and simulation of artificial societies.

JOHN A. BACHMAN is a Scientist at Charles River Analytics. Mr. Bachman’s research interests include human behavior representations, machine learning and data mining algorithms, and the simulation of complex systems using cellular automata. He graduated with a B.A. in Computer Science and Music from Harvard
University in 2002, with additional coursework in the University’s interdisciplinary Mind, Brain and Behavior program. For his senior thesis he developed “SoundSpace,” a software prototype demonstrating the possibilities of immersive and interactive virtual sound environments.

KAREN A. HARPER is a Principal Scientist at Charles River Analytics. Ms. Harper’s major research interests at Charles River Analytics are focused in the development of Human Behavior Representations (HBRs) modeling the human decision-making process in high-stress, time-critical situations. Ms. Harper is responsible for the continuing development and application of Charles River Analytics’ agent-based model of human behavior known as SAMPLE. Before joining Charles River Analytics, Ms. Harper obtained her M.S. from the Department of Aerospace and Mechanical Engineering at Boston University.

EVA HUDLICKA is a Principal Scientist and President of Psychometrix Associates, Blacksburg, VA. Her research interests include cognitive modeling, affective computing, decision support system design, and human-computer interaction. She received her BS in Biochemistry from Virginia Tech, her MS in Computer Science from The Ohio State University, and her PhD in Computer Science from the University of Massachusetts-Amherst. Prior to founding Psychometrix Associates in 1995, Dr. Hudlicka was a Senior Scientist at Bolt Beranek & Newman, Cambridge, MA.