Improving Trainee Engagement Levels through Adaptive Entity Behaviors

Joseph J. Sharp, Jason R. Potts,
Discovery Machine, Inc.
Williamsport, PA

jsharp@discoverymachine.com,
jpotts@discoverymachine.com

ABSTRACT

Successful training of individuals, whether in a live or virtual environment, depends heavily on producing an engaging experience which captures the attention of the trainee. The training system must not only replicate some real life environment, but must also provide trainees with a mentally stimulating experience. However, maximizing user engagement involves walking a fine line between boredom and cognitive overload. Further, this line shifts from user to user. An experience that is effective for one trainee may be completely overwhelming or possibly boring to another. To address this variation in user skill levels, the training environment should be able to recognize and adapt to the skill level of each trainee, while still adhering to the training goals of the simulation.

This paper presents work conducted under the Office of Naval Research, combining a user-friendly behavior modeling framework with biometric sensor feedback in order to create simulated entities who adjust their behaviors in response to a trainee's engagement level. This ability for entities within the simulation to adapt their behaviors based on the trainee's physical and mental state provides a mechanism by which the training scenario can constantly adjust itself to maximize its effectiveness. The recognition of user skill level is attained through an evaluation of a trainee's brain state, actions, and decision making in a scenario. This evaluation reports normalized values such as workload and engagement which are used to affect the simulation.

To provide an appropriate adjustment of the user experience, non-player characters in the simulation modify their behaviors to accommodate the current trainee. These behaviors are represented via a graphical task decomposition hierarchy, allowing non-programmers to define the appropriate actions for the entities to perform at any time, taking into account both the situation existing in the virtual environment as well as the engagement level of the player.

ABOUT THE AUTHORS

Joseph J. Sharp is a Software Engineer employed by Discovery Machine, Inc. He is a 2009 summa cum-laude graduate of Bloomsburg University of Pennsylvania where he studied and earned a Bachelor of Science degree in computer science. During his time at Discovery Machine, Joe has been involved in various behavior modeling projects, where he has played a significant role in developing technologies to aid in the process of using Discovery Machine tools to model complex human behaviors.

Jason R. Potts is the Director of Intelligent Systems for Discovery Machine Inc., overseeing the software development of their core technology offerings, as well as the application of their technology to customer problems across a variety of domains. Since graduating with honors from WPI with a Bachelor's of Science degree in Computer Science, Mr. Potts has been designing and developing software for Discovery Machine Inc. since 2001. This work has focused on development of tools facilitating knowledge capture, and representation of human expertise in a machine executable form. In addition to behavior modeling of simulated entities within constructive military training environments, these applications also include representation and automation of business process intelligence within commercial sectors such as manufacturing or pharmaceuticals.

Improving Trainee Engagement Levels through Adaptive Entity Behaviors

Joseph J. Sharp, Jason R. Potts,

Discovery Machine, Inc.

Williamsport, PA

jsharp@discoverymachine.com,
jpotts@discoverymachine.com

INTRODUCTION

With the increased interest in using virtual training environments in military training exercises (Cacas, 2011) come new challenges that test the ability of trainers to provide positive and effective training. Aspects of live training that are taken for granted become difficult to manage in the transfer to the virtual world.

When participating in live training events, trainees interact with live actors, and are observed by live instructors. Due to the nature of human communication, much of this interaction involves the trainers observing and interpreting trainee non-verbal communication. Without delving too deeply into the cognitive science that explains this ability, we as humans can generally gauge mental states such as confusion, deep thought, and anger based on vocal qualities, body language and facial expressions. (Knapp & Hall, 2009) This starts to become an issue when we take the trainee out of a live event and place them in front of a computer screen with a mouse and keyboard. How do we gain this same awareness of the trainee's mental state if instead of interacting with a human, the trainee is interacting with a computer screen?

There is significant research being done in an effort to give computer systems the ability to analyze voice characteristics (Huttunen et al., 2011) and facial expressions (Sarode & Bhatia, 2010) to accomplish these sorts of goals. However the intent of our system is not to emulate human non-verbal communication, but rather to gain an awareness of the mental states that non-verbal communication indicates.

An alternative path to this same awareness is a mechanism that gives the computer system a different window into this same information; an electroencephalography (EEG) headset which performs real-time analysis of EEG signals acquired from the wearer. Our partner in this project, Advanced Brain Monitoring, has performed extensive research in this area and has developed a solution that provides accurate brain state characteristics such as workload, engagement, drowsiness, and distraction. (Poythress et al., 2006) This allows us to circumvent the non-verbal communication issue without having to deal with the complexities involved in analyzing body language.

Once this awareness is developed, we must decide how to react to it, in order to provide the best training possible for the trainee in question. In a live event, an instructor may adjust training difficulty based on observed workload on the trainee, or live actors may adjust their level of cooperation to provide more or less of a challenge to the trainee. These kinds of adjustments are completed in an attempt to keep the trainee engaged in the training and prevent them from disengaging due to overload or boredom.

The process of matching the challenge of an experience to the abilities of the participants is described by Csikszentmihalyi (1990) in his work on the concept of flow. In this work, he describes how human behavior, enjoyment and engagement are highly affected by the nature of our experiences. He uses the term "optimal experience" to generally define experiences from which we derive great joy and view as highpoints in our lives. He argues that these experiences are realized through the voluntary maximum application of an individual's mental and physical skill to work towards the successful achievement of some task or goal.

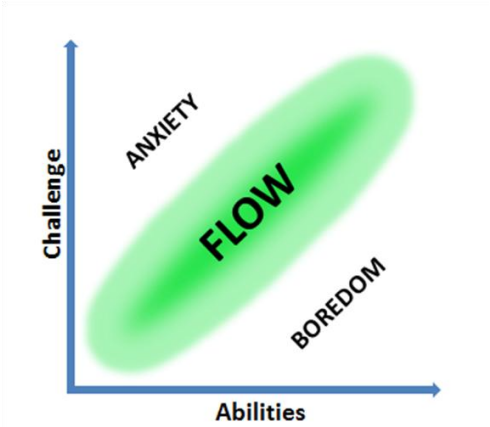

Figure 1. Csikszentmihalyi's Concept of Flow.

As depicted in Figure 1, experiences that are too challenging and seem impossible to accomplish cause feelings of anxiety and lead to eventual disengagement from said experiences. On the other hand, experiences

that do not present enough of a challenge quickly develop feelings of boredom, which also leads to disengagement. The goal is to match the challenge of an experience to the skill of the participant. When structured correctly, with well defined, achievable goals and explicit rewards both of which match participant skill levels, experiences can achieve flow. This elevates the experience for the participant and maximizes both enjoyment and utility of the experience. (Csikszentmihalyi, 1990)

When applied to virtual training and gaming, the same critical components of Csikszentmihalyi's concept of flow and optimal experiences apply. Games that allow users to achieve flow will cause users to enter a state of full cognitive engagement, to the point where time and external stimuli almost cease to exist. While it may seem unlikely that we will be able to achieve this extreme level of engagement in training simulations, we argue that the application of Csikszentmihalyi's concepts in a training simulation will inherently increase engagement for trainees in that simulation.

Our proposal then, is that a system that can make appropriate adjustments to accommodate trainee skill level would have to be able to perform the same tasks required of a live instructor or actor. It must acquire the same awareness of trainee workload and engagement, and then have the knowledge of how to adjust the training to appropriately maintain a level of difficulty that facilitates a state of flow for the trainee.

In this paper, we describe work conducted under the Office of Naval Research with the goal of alleviating the difficulties associated with the analysis and response to trainee cognitive workload in a simulated environment. This is achieved through the development of a system that enables rapid development of user-friendly character behavior models that, through expert knowledge extraction, are able to appropriately adjust their own behavior based on assessed trainee engagement level.

This system combines Task-Method-Knowledge (TMK) behavior hierarchies that represent virtual training character behavior and biometric sensor feedback to help gauge trainee workload. The behavior hierarchies are developed using a methodology which leverages subject matter expert (SME) knowledge to produce behavior models that appropriately reflect SME decision making and thought processes. Assessing trainee skill level and engagement is facilitated by monitoring the trainee's actions, decision making and cognitive workload throughout the scenario. Behaviors were developed for and deployed in the commercially available simulation environment Virtual Battlespace 2 (VBS2).

The adaptive nature of these entity behaviors is realized through two important mechanisms: behavior modification by instructors prior to the start of a training simulation, and real-time behavior modification by an automated game controller, termed the observer controller (OC).

Behavior modification prior to a training activity is akin to the pre-game instructions that would be given to live actors in a live training event. The instructor may give the actors guidance on the general skill level of the trainees entering the event, and in general provide some direction on how drastically they should respond to certain actions taken by the trainees. In this system, the instructor is able to tweak TMK behavior models to accomplish the same result.

Real-time behavior modification correlates to adjustments that live actors and instructors may make during a live training event based on the performance of the participating trainees. The training characters will react to trainee actions based on trainee engagement levels as dictated by pre-training settings, and will also follow instructions given to them by the simulated training instructor, the observer controller.

As a proof of concept for how this system is applied, we developed a scenario for VBS2 in which a trainee is able to enter an Iraqi village and participate in complex, realistic character interaction. The 40 characters that were developed for this scenario encompass many aspects of a typical Iraqi village, including religious leaders, law enforcement, and street vendors, as well as characters that act as the general population. These characters are based on character role information extracted from a database typically used for organizing live training events in locations such as the National Training Center in Fort Irwin, California.

The trainee enters this village with the objective of investigating a report of an individual who may be housing insurgents in the village. To successfully complete the objective, the trainee must locate characters in the village that have information regarding this and attempt to extract that information from them. Through this process, they must identify any characters that are suspected of supporting the insurgency.

Characters in this scenario will respond to direct interactions with the player, as well as to the workload and assessed skill level of the player to facilitate effective training.

TECHNOLOGY OVERVIEW

To provide a full understanding of the adaptive nature of the behaviors developed using this approach, we will have to first develop an understanding of the core technologies in play.

TMK Behavior Modeling

The Task-Method-Knowledge approach that we have developed for behavior modeling leverages visual programming elements that are assembled in a hierarchical fashion to represent and execute complex process models. Despite visual similarities to finite state machines (FSMs), this approach is in fact quite dissimilar to a FSM. Rather than focusing on states and state transitions, TMK focuses on logical processes that accomplish the task at hand. Research has shown that human behavior can be easier to represent in the form of a visual hierarchical task network than other traditional behavior modeling mechanisms, and the result facilitates natural process comprehension. (Hauser, 2010)

This approach is based off of work completed by Chandrasekaran (1986) in which he proposes that patterns of process knowledge, termed "generic tasks", as well as traditionally accepted concept knowledge, can be used to represent human behavior. Our TMK hierarchies are representations of process and concept knowledge that can be applied to solve specific tasks.

This concept of a task decomposition hierarchy is one that can be followed quite naturally by experts, who are generally the individuals that describe the kinds of processes represented in TMK. This leads to one of the main differences and benefits of this approach when compared to traditional behavior modeling approaches: the knowledge acquisition mechanism. This methodology guides the capture of process knowledge directly from experts using visual TMK elements. This visual representation facilitates SME participation at all levels of behavior development, from conceptualization to deployment.

This visual paradigm provides increased contextual awareness for a SME and allows for better understanding of the overall system when compared to a rules based, lisp style representation. With this increased understanding comes a more effective and efficient elicitation of expertise. The SME benefits in the visual representation of their knowledge, in that they gain an increased understanding of processes that may have previously been performed on an entirely subconscious level. Keeping the SME in the loop throughout the assembly of process models promotes accuracy and depth of process knowledge. Experts can quickly point out areas of knowledge deficiencies as well as incorrect assembly of process knowledge components.

The core elements of our implementation of the TMK language are hierarchical visual representations of the various process knowledge components: tasks, methods, and procedures. Tasks represent things that need to be done, and methods and procedures are ways of accomplishing those tasks. Methods represent more complex ways of accomplishing tasks, and break down into subtask groups which are themselves made up of individual tasks. Procedures are simple action oriented components that cannot be further decomposed and thus act as the leaf nodes of a TMK hierarchy. An example TMK structure can be seen in Figure 2.

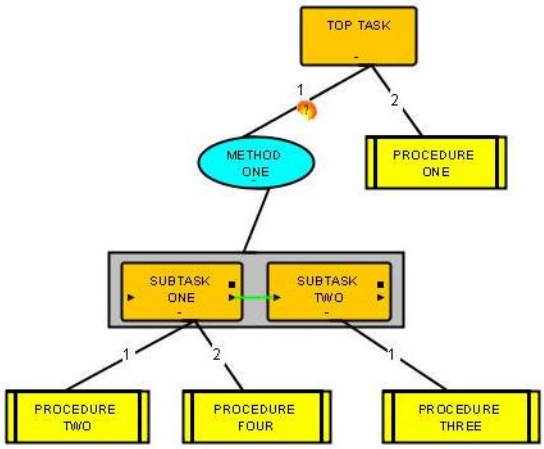

Figure 2. Example TMK Model.

A task can have multiple ways in which it can be accomplished, and thus can have multiple method and/or procedure "child" elements. These child elements, are executed in a user specified order, and under user specified conditions, to reflect the decision making that takes place in determining how to complete any real world task.

An example of this structure would be a task such as "Eat Lunch." This high level task signifies that the goal of this hierarchy is going to be consuming food for lunch. This could be done in several ways, depending on the situation: "Order Out", "Go to Restaurant", and "Eat Packed Lunch". Under different circumstances, any of these mechanisms could be used to accomplish our goal. If there is a business client on site, we may go to a restaurant. If we have lunch packed, we will probably eat that unless we packed a boring lunch and have since decided that takeout is the better solution.
In this scenario, the "Go to Restaurant" solution would be best represented as a method, since it is a fairly complex activity, and would break down into subtasks

such as "Choose Restaurant", "Travel to Restaurant", "Order Food", etc. Each of these subtasks would then have their own sub-hierarchy to accomplish the current goal at hand.

The "Eat Packed Lunch" solution may be represented as a procedure, since it is quite simple and has few if any decision points.

Due to the specific nature of the rule definition for TMK element decisions, the rules themselves are generally very easy to express. People are quite adept at describing conditions at this level. For example, consider the question: "What are the circumstances under which you would order take out versus eat a packed lunch?" While this is a rather simple example scenario, it demonstrates the level at which rules are developed in TMK. Experts don't have to define general rule sets that encompass all aspects of a situation; they simply define specific rules for specific decision points in a hierarchy, which provides a concrete context for rules, therefore simplifying the process. As such, every decision point in the hierarchy is an opportunity for experts to inject human aspects into how that hierarchy is executed.

Behavior modeling using visual elements in this fashion provides a natural mechanism for authors to represent tasks that they are trying to encode, as evidenced by Hauser (2010). Given this ease of understanding, we can effectively employ the instructional participants of a live training scenario to act as experts for developing the various aspects of a training experience.

Brain State Monitoring

The analysis of trainee cognitive workload is accomplished by interfacing with an electroencephalography (EEG) headset (Figure 3) which performs real-time analysis of EEG signals acquired from the wearer. This hardware and accompanying software, provided by our partner Advanced Brain Monitoring, Inc. (ABM), produce operational real-time monitoring of brain state characteristics such as alertness, attention, distraction, engagement and mental workload. (Berka et al., 2010)

Figure 3. B-Alert X10 Headset

The functionality provided by this wireless headset has been utilized in various applications, from sleep optimization, and fatigue monitoring to psychophysiological profiling of individuals during training to identify areas of improvement in novice trainees.

When interfaced into our closed loop, real-time simulation environment, brain state characteristics are normalized and made available to executing behavior models. This data collection and normalization mechanism is one of the core offerings of the software provided by ABM. Through extensive user testing and analysis, a normalization algorithm has been developed that can effectively approximate human workload and engagement characteristics based off of the EEG readings taken from the headset. (Johnson et al., 2011)

These values allow behavior models to develop a sense of brain state trends that may occur in a trainee to indicate either cognitive overload or boredom. Behavior models are then able to use these values to affect their decision making throughout a simulation. The subject matter expert that is assisting in the definition of these behavior models can indicate decision points in the model that should account for the cognitive workload of the trainee.

Virtual Battlespace 2 Behavior Toolset

The core work completed in the project was done so in a customized TMK behavior authoring environment, developed for Virtual Battlespace 2 (VBS2). VBS2 is a well known and widely used virtual training solution capable of simulating a wide range of situations at the tactical level. It is a platform that is widely used by various branches of the U.S. military, including the U.S. Army and U.S. Marine Corps, for training, mission rehearsal and experimentation. This widespread use is the core reason that it was chosen as the platform for performing this work.

Using VBS2's plugin architecture and Application Scripting Interface (ASI), an interface was developed to build and attach TMK behavior models to characters in VBS2. This interface includes a behavior composition tool which enables behavior authoring as well as a behavior execution engine which interfaces with VBS2 at simulation runtime.

Behavior Composition

The authoring of behaviors is completed in a custom TMK authoring environment, shown in Figure 4, that has functional elements that are directly tied to VBS2 scripting functionality.

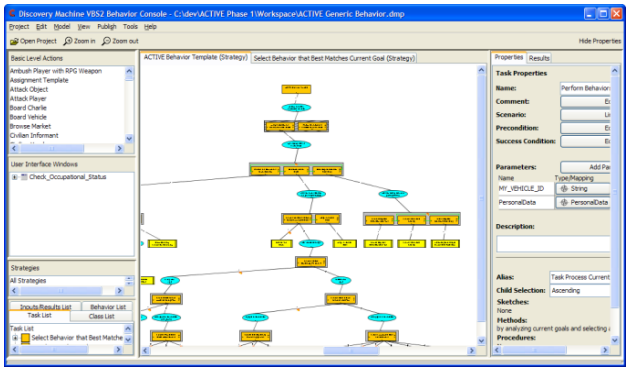

Figure 4. Discovery Machine VBS2 Behavior Console

In this tool, TMK hierarchies are built up to represent character behavior and can then be published as executable behaviors, called gears. Once a behavior has been published, it is immediately available for use in a simulation scenario, via the runtime component of the toolset. Users are provided with advisors that assist in leveraging basic behavior elements, called basic level actions (BLAs) to build up appropriate behavior models for their desired simulation characters (specifics will be discussed later in the paper).

Behavior Execution

To execute behaviors authored in the behavior console, users launch a TMK behavior engine along side of VBS2 at simulation runtime. This behavior engine uses the VBS2 plugin interface to initialize a runtime link to the simulation. This runtime link is used to post action and query commands to the simulation to affect character movement and actions and to retrieve vital data from the simulation.

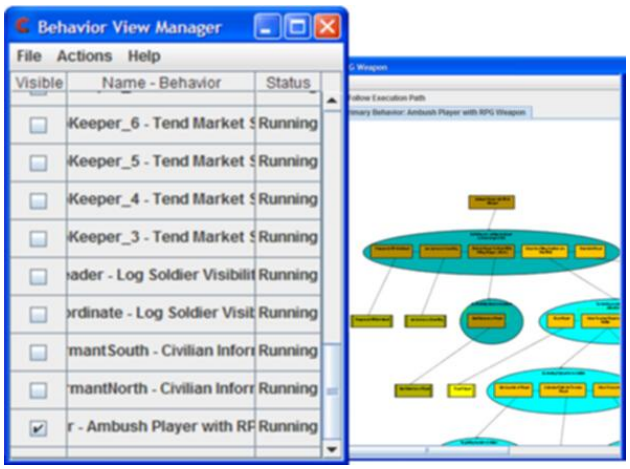

Figure 5. Runtime Behavior View Interface.

One of the core aspects of behavior execution is the viewing of behaviors at runtime. As behaviors are launched and attached to characters in a VBS2 scenario, those behaviors are displayed in an interface which allows users to view behavior models as they execute.

Figure 5 shows the view selection interface on the left, and an open, executing behavior on the right. This is useful for both debugging purposes while authoring behaviors, and reviewing purposes when observing behaviors during training. These execution views provide a window into the decisions that are made at runtime including real time data values and indications of execution paths taken by the executing model.

BEHAVIOR ADAPTATION

Pre-Training Adaptation

Behaviors that are built using this system can be modified at any point prior to the launching of the training simulation to facilitate highly specific training objectives in a scenario. Since all behaviors are defined in TMK behavior models, this modification can be completed with relative ease.

Users may want to inject specific event handling mechanisms to individual characters in a scenario. Users could inject mechanisms for responding to commands given to them by other characters, or perhaps a specific response to some expected action by the player. Examples of this would be the modification of attitude thresholds that characters use to determine whether or not to fully cooperate with the trainee, or the addition of simplified interaction options to characters that are likely to interact with the player.

This ability to quickly modify existing behaviors provides a flexible authoring system that allows users to effectively develop behaviors at various levels of complexity to achieve their desired training objectives.

Real-Time Adaptation

In order to maximize flow for the trainee during the actual execution of the simulation, two primary mechanisms were incorporated which enable characters to adapt their behaviors in response to biometric data provided by ABM's hardware. Both of these techniques center on an automated "observer-controller" (OC) behavior model which exists as a TMK behavior attached to an invisible entity within our training scenario. When the player's biometric data indicates that his or her engagement levels have fallen out of the target zone, the

OC responds by sending communications to the characters within the training simulation which cause them to either modify their mood and attitude toward the trainee, or take specific action to guide the player toward his or her goal.
The first technique involves communicating with the entire population of the game world, and either increasing or decreasing their mood and attitude. This has the effect of reducing (or possibly increasing, in situations where the player requires a additional challenge) the overall hostility level experienced by the player. For instance, if the OC improves the population's mood, those characters will be less likely to engage in activities which would increase the player's stress level, such as protesting. If the OC increases the attitude of the population towards the player, the player will have more conversation options available to him, making it easier to extract information which is vital to his mission. Through these effects, the system is able to reduce (or increase) the player's anxiety and workload by making the scenario more or less challenging.

In addition to modifying the mental state of the characters within the simulation, the OC can also direct specific characters to perform specific actions, designed to guide trainee and get him "back on track" in situations where the trainee may have become lost. For instance, the OC may cause an important character to move to the area where the player is located, making it more likely that the player will interact with them. If the player continues to ignore the character, the OC may direct the character to begin engaging in activities which are more likely to draw the player's attention, such as arguing with another character in front of the player. In situations where the player continuously fails to respond to the character, the OC may finally just direct the character to take the initiative and proactively start up a conversation with the player.

In these ways, the automated OC functions similar to the director of a movie, choreographing the activities of the characters in order to keep the player engaged. Further, since the OC is also represented as a TMK model, it offers the same benefits to end users that the character models do. Namely, its execution can be viewed at runtime, in a graphical and human readable form, allowing the trainer to understand when the player's engagement level fell, and what actions the OC performed in response. Finally, because the OC model has access to the same extensive capabilities in terms of integration with various external systems, it can even choose to directly log these interventions for later review by trainer and trainee alike, improving the after action review of the training event.

ADAPTIVE BEHAVIOR MODELING

To facilitate adaptive character behavior authoring in our simulated entities, the TMK authoring console was used to develop a general behavior template that all characters use as a base for their behaviors. This template is based loosely off of work funded by NAVAIR-TSD and the Office of Naval Research, which demonstrates a customized behavior modeling framework for specification of constructive entities within the Joint Semi-Automated Forces (JSAF) simulation environment. (Potts, Griffith, Sharp & Allison 2010)

The creation of this template behavior provides a starting point that naturally facilitates adaptive behavior authoring and modification. As previously described, one of the core aspects of program control in TMK is creation of the decision points in a hierarchy that determine in what manner a certain task should be completed, or decide when that task has achieved completion. By introducing a goal directed architecture to control entities that is based on human behavioral characteristics, we have exposed certain aspects of human decision making that can be modified with relative ease by end users of this system. This ease of modification encapsulates the first component of behavior adaptation which is that instructors and scenario builders are able to comprehend and modify character behavior models in a time frame that facilitates new behavior creation and modification for individual training groups and trainees.

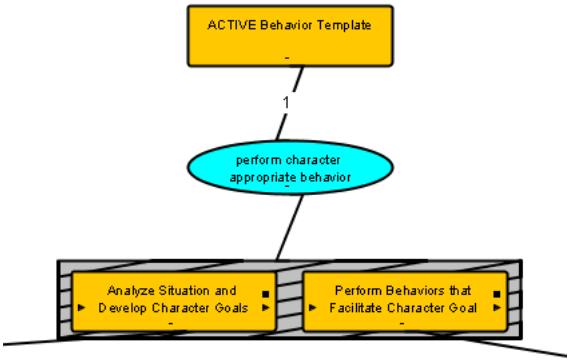

Figure 6. Top-Level Behavior Template.

The template itself lays out a goal oriented behavior structure that constantly analyzes the current situation and develops appropriate character goals based on that analysis. To accomplish this, the template has two main concurrent tasks that execute simultaneously to affect entity behavior. The first task is “Analyze Situation and Develop Character Goals”, and the second is “Perform Behaviors that Facilitate Character Goal Achievement” as seen in Figure 6.

Situation Analysis and Goal Development

In the situation analysis and goal development branch, all sensory input of the character is processed and used to build up an understanding of the current situation. This situational awareness is used to develop appropriate goals for the given character.

In the development of this awareness, we examine simulated physiological signals such as hunger, fatigue and critical health, as well as physical sensory input, specifically auditory and visual. These low level signals provide a general comprehension of the factors that have the highest impact on the general character behavior that we are interested in. This information helps develop an accurate mental image of what is actually occurring in the current situation and is used to develop reasonable goals.

To make the goal assessment and achievement architecture functional, we developed a set of character goals that represents any high level goal that one of our characters is able to perform. These are as follows: "Perform Occupational Duties", "Eat", "Sleep", "Initiate Conversation", "Engage in Conversation", "Take Cover", "Interact with Multi-National Forces", and "Idle".

When completing character goal assessment, we take into account all of the information that was analyzed about the situation, including general information about the state of the world such as the time of day. The goal analysis is completed in a hierarchical manner based on the priorities of the character in question. We assume that the foremost goal of any character in our non-kinetic scenario is survival, so the first thing that gets checked is critical health. Critical health is a representation of damage taken due to kinetic events such as bullet or shrapnel damage. If we recognize that we have taken any sort of critical damage or are made aware of anyone in our general vicinity that has taken critical damage, our goal is set to "Take Cover". The next factor that we take into account is whether or not we have an occupation, and if so, whether or not we are supposed to be at work. If we are supposed to be working, we are less likely to engage in social conversations, or stop what we are doing to eat or sleep. This affects our weighting in deciding these types of things. For instance, if I am not working and I am a little hungry, I might swing by the market and grab something to eat. However if I am working, I won't stop to eat until I am quite hungry and able to take a break for lunch. If we are not working, we will eat, socialize, and perform other idle behaviors at will. Eventually we will get tired and decide to sleep. These kinds of decisions are affected by our mood and proximity to known associates, as well as our hunger and fatigue. We might decide to go start up a conversation with a social associate especially if mood is very low and want to express frustrations.

Character Goal Achievement

Each goal that the character can work towards is represented by a method under the goal achievement task as seen in Figure 7. This task exists in a loop that repeatedly checks the current goal, as formulated in goal assessment hierarchy, and tasks itself with the appropriate method for accomplishing that goal. For example, if the goal assessment hierarchy decides that "Eat" is the current goal, the goal achievement hierarchy will gracefully abort its current activity, and switch to the "Eat" method.

Each of these methods is made up of groups of subtasks, which are defined as BLAs that complete the goal that is specified by the method. The "Eat" method could be made up of BLAs such as "Locate closest Deli" and "Go to and Eat at Deli", which would cause the character to eat a meal at the closest deli to his or her current location.

Character Mental States

Throughout the scenario, character behavior will vary significantly based of their own simulated mental state. Each character has two values that are of interest: mood and attitude towards the player. Independent modification of these values has a direct effect on the actions that each character takes. If a character's mood is quite low, in other words quite angry, and they have a low

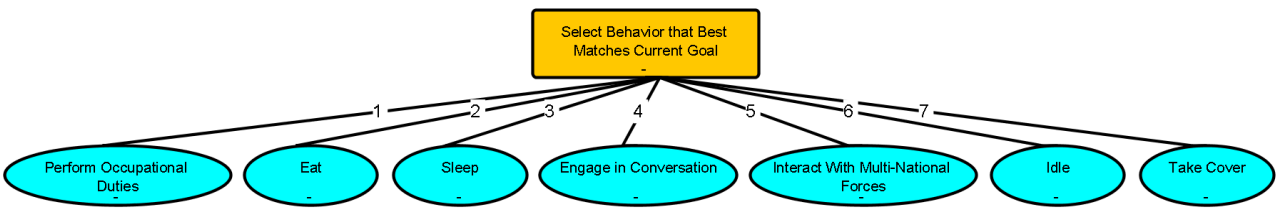

Figure 7: Goal Achievement Task and Methods.

attitude towards the player they may form a protest. However if their attitude towards the player is high, they may be inclined to approach the player and discuss their issues for a possible positive resolution. The mood mainly affects how likely a character is to seek out interaction with the player and other characters, while the attitude towards the player affects how likely each character is to cooperate and provide useful information to the player.

These values are affected by each character's experience in the simulation. Characters influence each others' moods and attitudes based off of simulated verbal communication, which results in attitude and mood propagation throughout characters in the simulation. This mood propagation is completed through the simulation of a natural flow of information through the social network model represented in the character database. Each character will have social, family and business associates, all of whom the player is likely to interact with and disseminate information to. Through these interactions, mood and attitude will spread through the village.

These moods and attitudes are ultimately affected by interactions with the player. If the player is behaving in such a way as to be highly offensive, mood and attitude will both be diminished. The player has the opportunity to approach and attempt to engage in communication through an in-game chat interface, with any of the characters in the simulation. Through this communication the player has the ability to directly affect these characters as described in the interaction modeling. Player actions also indirectly affect all nearby characters, as these characters observe player behavior that is occurring even when they are not directly interacting with the player.

For instance take any given character, maybe a construction worker. This character at some point is approached by the player and has very negative interaction with the player. Assuming this character has a social association with a local street vendor, he is likely to stop by and converse with this vender the next time he visits the market place. Some portion of the negative feelings that the construction worker has developed towards the player will be influenced onto the vendor. All future interactions that other characters have with these two will result in further attitude dispersion.

Building Basic Character Behaviors

Given this general behavior template, we extract character data from a given character database and use that data to develop the template into operationalized behavior models for use in a VBS2 scenario. To formalize this template into an operationalized behavior model, we use an advisor system which guides us through choosing characters, classifying character data, and choosing relevant activities for that character to complete.

Character data is extracted from character databases typically used for organizing live characters in a live training event. This information includes characteristics such as occupation, level of education, social, family and business associates, as well as general feelings towards U.S. forces, religion, and military training. The advisor interface presents this list of characters to the user, who selects a single character, and is presented with data for that character.

The advisor automatically processes much of the information, simply mapping relevant information in the database to corresponding behavior characteristics stored in the character models. The advisor elicits user input for things that would require heavy processing or complete guessing on the advisor's part. Since most of these databases were intended for human processing, certain fields are not machine friendly and may contain typographical errors.

For example, a construction worker's occupation may be listed as "Construction", "Construction Worker", "Construction Foreman" or "ConstructionWorker", among other things. The user is asked to categorize occupation into one of several categories including: Professional, Laborer, Unemployed, and Clergy. This provides vital information to the advisor for determining important behavioral characteristics for this individual, such as typical work hours and activities.

After, several such prompts, including the selection of work and socially related basic level actions for use in various parts of the model, the advisor completes the assembly of a fully functional behavior model for the character in question.

Character Interaction Modeling

The critical step in making behavior models that can appropriately interact with a trainee is the assembly of each behavior model's interaction model. These interaction models are what determine how each specific character is able to interact with the player. The behavior author is provided with an interface to define the knowledge that each character has in his or her arsenal for communication with the player. The core aspects of the knowledge definition are the definition of questions and correlating answers that could be asked to retrieve

this knowledge from the character, the definition of knowledge and answer privacy level, and the mood and attitude effect that results from the actual asking of specific questions.

The answer given to any particular question depends on several factors including the privacy level of the knowledge and the relationship of the individuals engaged in conversation.For example, consider Khakid Kalabat, a farmer in our simulated village of Medina Wasikh. His character definition in the character database contains information about his life, family, and personal history including a section of general character motivation background as follows:

"You are a farmer and your family has always been farmers. You are hoping to find work in Medina Irwin. You are married and have four children; two sons, Abdul and Rashid, and two daughters, Ahd and Dahab, your wife's name is Alia. Your son Abdul was a Sergeant in the Iraqi military. You have not heard from Abdul for two years now. You fear he may be dead, but do not discuss it with anyone but your wife. You have have been married for 24 years. In 1992 you were imprisoned by the regime for 6 months; you have not been able to talk anyone about your experinces there. Your remaining family has followed you to Medina Wasikh when the Iraqi military and insurgents entered Mahmudiya. In your spare time you enjoy smoking and drinking tea with your people and talking about the great changes ahead for your country. You only have a minimal amount money. Members of the town occasionally provide you with money to buy food with. While you are a Shia Muslim, you are not an extremely religious person. You follow your religion, but are not defined by it. You do answer calls to prayer."

As a user, I will want to define questions that allow access to this information as I see fit to support my training goals. One such topic for questions would be Khakid's feelings towards the old Saddam regime and Saddam's removal from power. I define question phrasing for how this question would be asked directly to Khakid, as well as how this question would be asked about Khakid for the case in which the player is asking another character in the simulation about this topic.

Asked to him: "Are you happy that the Saddam regime has been removed from power?"

Asked about him: "What are Khakid Kalabat's feelings towards the old regime?"

I would proceed in this process to specify the various answers to these questions. The specifics of these answers depend on the situation in which the questions are asked, and to whom they were asked. They depend on the asked character's relationship to the asker, the level of privacy of the knowledge, and the asked character's current mood and attitude.

Consider the question directly asked to Khakid from a complete stranger. Khakid depending on his mood, may respond with "Yes, it is better now.", It is fine.", or "I am very busy, please don't bother me with such questions." If this question were asked by a close companion, he may respond with more details or enthusiasm: "Yes, we are so happy to be free of the old regime. Our family is very glad that we are no longer oppressed by Saddam." If this question were asked of another character about Khakid, the answer would depend on that character's relationship with Khakid, as well as with the player. If the character was a close companion of both Khakid and the player, he or she may divulge information about the absence of Abdul, Khakid's son as a source of anguish caused by the old regime.

Throughout the trainee's experience in the simulation, questions such as this are presented in an in game interaction interface as he or she approaches and converses with characters. The user selects questions out of the ones presented and to ask to the character, which is the basis of all character interaction in the scenario.

After an appropriate level of interaction modeling has been completed, the fully interactive nature of this scenario can be realized. The level of effort required and complexity of said modeling is completely dependent on the training goals of a specific scenario.

CONCLUSION

The end result of this work is a system that we believe demonstrates a unique combination of visual behavior modeling and real-time user statistics that facilitates the development of highly adaptable entity behaviors. Using this combination of technologies alleviates some of the more difficult aspects of traditional behavior development. This allows SMEs to leverage complex biometric technology to effectively dictate how simulated entities will provide effective simulated training to trainees with varying levels of incoming skill.

As this architecture matures, Discovery Machine will continue their efforts in validating the use of visual TMK hierarchies for this type of application. Our current efforts include expanding the domain of expert knowledge that we have at our disposal for use in this system.

Working with other major players in the modeling and simulation community will give us the opportunity to tune this architecture to best match the major work being completed by the community at large.

ACKNOWLEDGEMENTS

The efforts described in this paper have been sponsored by the Office of Naval Research and supported by our partner Advanced Brain Monitoring.

REFERENCES

Berka, C., Pojman, N., Trejo, J., Coyne, J., Cole, A., Fidopiastis, C. Nicholson, D. (2010) Neurogaming: merging cognitive neuroscience & virtual simulation in an interactive training platform. *Proceedings of the 1st international conference on neuroergonomicsat the 3rd international applied human factors and ergonomics.*\

Cacas, M. (2011, June). Training hits the virtual target. *SIGNAL Magazine*, Retrieved from http://www.afcea.org/signal/articles/templates/Signal_Article_Template.asp?articleid=2629&zoneid=318

Chandrasekaren, B. (1986) Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design, *IEEE Expert*. Fall, 1986.

Csikszentmihalyi, M. (1990). *Flow: the psychology of optimal experience*. New York, NY: Harper Perennial.

Hauser, R. (2010). Analysis and transformation of behavioral models containing overlapping patterns. *Journal of Object Technology, 9*(3), Retrieved from http://www.jot.fm/issues/issue_2010_05/article4.pdf

Huttenen, K., Keranen, H., Vayrynen, E., Paakkonen, R., & Leino, T. (2011). Effect of cognitive load on speech prosody in aviation: evidence from military simulator flights. *Applied Ergonomics*, *42*(2), Retrieved from http://www.sciencedirect.com/science/article/pii/S0003687010001195 doi:10.1016/j.apergo.2010.08.005

Lee-Urban, S. (2005). *TMK Models to HTNs: Translating Process Models into Hierarchical Task Networks* (Master's thesis). Retrieved from http://www.lehigh.edu/~sml3/publs/StephenUrbanMSThesis.pdf

Johnson, R., Popovic, D., Olmstead, R., Stikic, M., Levendowski, D., Berka, C., (2011) Drowsiness determination through EEG: development and validation. *Biological Psychology in press.*

Knapp, M., & Hall, J. (2009). *Nonverbal communication in human interaction*. Boston, MA: Wadswroth | Cengage Learning.

Poythress, M., Russel, C. Siegel, S., Tremoulet, P., Craven, P., Berka, C., Levendowski, D., Chang, D., Baskin, A., Champney, R., Hale, K., & Milham, L. (2006) Correlation between expected workload and eeg indices of cognitive workload and task engagement. *Proceedings of the 50th annual meeting of the human factors and ergonomics society.*

Potts, J., Griffith, T., Sharp, J., & Allison, D. (2010). Subject matter expert-driven behavior modeling within simulations. *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2010*

Sarode, N., & Bhatia, S. (2010). Facial expression recognition. *International Journal on Computer Science and Engineering*, *2*(5), 1552-1557.

Silverman, B., Johns, M., O'Brien, K., Weaver, R., & Cornwell, J. (2002). Constructing virtual asymmetric opponents from data and models in the literature: case of crowd rioting. *Proceedings of the Conference on computer generated forces and behavioral representation*, http://repository.upenn.edu/cgi/viewcontent.cgi?article=1001&context=hms

Discovery Machine, Inc. Patents Awarded:
US Patent 7,257,455 B1 – System and Method for Collecting and Representing Knowledge
US Patent 7,757,220 B2 – Computer Interchange of Knowledge Hierarchies
US Patent 7,958,073 – Software and Methods for Task Method Hierarchies

Discovery Machine, Inc. Patents Pending:
US Application 11/781,671 – Reflective Processing of Computer Hierarchies