ABSTRACT: A current technical shortfall and, therefore, a priority for computer generated actor (CGA) development is the appropriate portrayal of realistic behaviors. The CGA behavioral problem is primarily the result of the lack of appropriate CGA knowledge representations, decision mechanisms, and software architectures to support human behavioral representational processing. Incorporation of user modeling techniques into CGA behavioral modeling efforts can provide the CGA with the necessary flexibility to respond to a given environmental state in a manner that more closely mirrors real-world human behaviors. As a result, the realism of a distributed virtual environment utilized for training or mission rehearsal is increased. User modeling can be effectively utilized for CGA development by applying this approach in a way that treats human and computer actors the same in the environment, makes explicit the reasoning about the purpose of system adaptations and decisions, takes into account actor motivation and emotions, and presents a unified model of collaborative, cooperative, and adverse behavior.

1. Introduction

A pervasive problem for computer generated actors (CGAs) is the portrayal of realistic behaviors within a “synthetic battlespace” distributed virtual environment (DVE) venue used by the military for training and mission rehearsal. This problem arises primarily from the inappropriate knowledge representations, decision mechanisms, and software architectures for current human behavioral representation and processing within a CGA. By employing user modeling techniques in CGA behavioral model development and execution, a CGA can respond to a given environmental state in a manner that more closely mirrors real-world circumstances and, therefore, increases the realism of the DVE.

Our research addresses realistic CGA behavioral portrayal by the incorporation of a user modeling approach into the decision processing of our generic CGA software architecture long-term decision engine. The knowledge representation for our user model applies a unification-based grammar framework. Within the framework, a complex data structure referred to as a feature structure encodes information from various sources as a declarative specification of knowledge. The knowledge represents both the collection of knowledge about the CGA and the use of the CGA platform within the environment. The employment of a user model within the long-term decision engine of our CGA enables CGA behaviors that mirror those behaviors a traditional operator would normally perform for overall system control and utilization.

The next section briefly describes our generic CGA architecture and its approach to decision making and then introduces user modeling and motivates its use for CGA decision processing. Section three presents our user modeling strategy of unification-based grammars and feature structures. Section four discusses our application of user modeling to decision processing within our CGA long-term decision engine and the benefits of using this approach. We conclude the paper with a summary of our current development status and recommend avenues for future research.

2. Background

This section describes information concerning our generic CGA architecture and the area of user modeling. The third subsection addresses the motivation for the incorporation of user models into a CGA decision processing strategy.

2.1 CGA Architecture Overview

Our development goal for a general CGA architecture was to provide an architecture for CGAs that naturally accounts for “variety” in the observable performance of a given type of CGA as well as supporting the assembly of vastly different CGAs such as tanks and aircraft. Our architecture consists of highly modular components where we precisely define and minimize interdependencies. We model actor data processing in two stages, computations of the physical world state contained with the Physical Representation Component (PRC) and then reasoning upon the state, which occurs in the Cognitive Representation Component (CRC). The resultant reasoning outputs are used to control the actor and to generate outputs for the DVE.

The PRC encapsulates all the physical attributes and properties of each actor such as dynamics models, actor-specific properties, performance capabilities, weapons load, and sensors. The sensor interface portion of the PRC is composed of a set of entity specific units where each unit takes world state information and restricts it to simulate the quality of information that would be
available to the real world system. The PRC also computes physical state changes, such as the new CGA position in the virtual environment and the ballistic path of bullets. The CRC contains the intelligent decision making processes, the skill level representations, and the knowledge required by the CGA. The knowledge includes the overall mission, goals and objectives, plan generation, reaction time, and crisis management ability. The CRC has three reasoning engines, the Long-Term Decision Engine (LTDE), Mid-Term Decision Engine (MTDE), and the Critical Decision Engine (CDE), which are scoped to provide decisions along different focuses and timelines. Any form of reasoning can be used in the engines. The Arbitration Engine (AE) takes the output of the three decision engines and, utilizing a selection mechanism that incorporates a combat psychology model, skill levels, and other performance factors, selects a decision. The AE can simulate abstract concepts such as “fear” (by selecting the CDE's decision), “bravery” (by ignoring the CDE’s decision and choosing the MTDE's), and “indecisiveness” (by choosing to ignore all decisions).

### 2.2 User Modeling

Most experts agree that a computer “decision” in support of a user must be based on an accurate representation of the users’ knowledge and interactions with the system, called user models. The main function of user modeling is to determine what the user intends to do within a system environment (“user intent”) for the purpose of assisting the user. Nuten defines human (user) intent as “mental states which drive actions.” One approach to predicting user intent identifies the salient characteristics of the domain environment and specifically determines the goals a user is trying to achieve. This approach is based on the belief that what a user intends to do in an environment is the result of events occurring in the environment and the goals the user is trying to obtain as a reaction to stimuli. To achieve a goal, a user must perform certain actions.

The two user models of primary interest to the CGA community, behavioral and cognitive models, are performance models in that they are used to determine a user’s future actions. The primary difference between the two models lies in the level to which the user is modeled. Both models observe the user’s execution of actions; however, cognitive models attempt to determine the user’s goals, whereas the behavioral model directly forecasts user activity. Because of their predictive nature, both model types have been applied to adaptive computer systems.

Cognitive models have been studied by psychological researchers for many years. Cognitive psychologists theorize that humans form cognitive models of their environment to make sense of the information they observe. Analogously, a DVE actor may also use a cognitive model of its environment and its “operator” as it determines how to operate in that environment. A cognitive model represents the human user as a collection of goals and a set of actions to accomplish the goals. The cognitive model allows the intelligent agent to attempt to determine what the goals of a human user would be in a given situation. Once the agent has determined the user’s goals, it then locates a set of actions that will assist the user in accomplishing the goal. In a given situation, the agent must decide the goals to pursue and the methods used to achieve them.

The behavioral model, another performance type of user model, attempts to address some of the difficulties encountered when using a cognitive user model. In a behavioral model, the behavior of a system is manifested in input-output relationships; the user’s behavior can be defined as a succession of states. Put another way, a behavioral model represents the human user as a collection of sequences of actions that the user performs. The system does not attempt to determine the user’s goal, as in cognitive modeling, but directly predicts future user actions based on the status of the environment and past user actions.

### 2.3 Motivation for Utilization of User Modeling Approach

User models, both cognitive and behavioral, are becoming recognized as a foundational source of information for use within many types of systems utilized extensively in military DVEs for training or mission rehearsal: decision support, information retrieval and analysis, intelligent user interfaces, and intelligent software agents. These areas are all components of an effective CGA; where an effective CGA is one that can not be recognized as being computer controlled and, therefore, triggers appropriate human responses. With the current emphasis on incorporating realistic CGAs into simulation and training efforts within the Department of Defense modeling and simulation community [6], user modeling is becoming an increasingly significant area of interest.

User modeling techniques can be effectively utilized in CGA development by applying a user modeling approach that treats human and computer actors the same in the environment, makes explicit the reasoning about the purpose of system adaptations and decisions, takes into account actor motivation and emotions, and presents a unified model of collaborative, cooperative, and adverse behavior. In addition, employment of a user modeling approach allows the direct incorporation of knowledge derived from cognitive task analysis techniques into the knowledge processing structure of a CGA.

Within user modeling and specifically for its application to CGA behavioral modeling, a representation that has the flexibility and power to deal with the uncertain environment and the dynamics of modeling user goals and actions is required. Employing a knowledge representation that correctly captures and models uncertainty in human-computer interaction can improve the modeling of the user and, therefore, the CGA behavior based upon this user model. For these reasons, we developed a representation to capture user goals and actions in the environment through cognitive task analysis and then utilized this representation within our CGA.

---

*In user modeling, a distinction is made between competence models, which determine what a user could do, and performance models, which determine what a user is likely to do.*
reasoning architecture to produce human-like CGA behaviors.

3. Representation for User Model Based CGA Reasoning

Our proposed partial solution to the problem of incorporating user modeling techniques into CGA behavioral modeling is the representation of feature structures and its associated manipulation technique of unification.

3.1 Feature structures and unification

Unification grammars encode knowledge as a declarative specification of constraints rather than as a procedural definition of derivations. Within the unification-based grammar framework, a complex data structure referred to as a feature structure\(^3,7\) or functional description\(^2\) encodes information from various sources as feature/value pairs. A feature structure may be viewed as a partial function from features to their values\(^2\) and complex feature structures contain partial functions from features to values that are themselves feature structures. Frames\(^4\), a popular artificial intelligence representation, resemble feature structures but are more limited in scope. Feature structures also resemble first-order logic terms but have several important distinctions\(^3\). Most importantly, feature structures remove first order restrictions on order dependence and fixed arity.

Knight\(^3\), Shieber\(^7\), and Kay\(^2\) all discuss unification as the sole manipulation process for feature structures. The unification problem may be abstractly stated as follows: Given two descriptions \(x\) and \(y\), can we find an interpretation \(z\) that fits both descriptions? Unification of feature structures results in a new structure that contains all of the information from the two structures to be unified, assuming no attribute/value pairs contain conflicting information. Unification of feature structures can be defined with the introduction of the partial ordering relation \(\text{subsumption}\). Feature structure \(D\) \(\text{subsumes}\) feature structure \(D'\) if \(D\) contains a subset of the information in \(D'\).\(^7\) Then, \(\text{unification}\) of feature structures \(D'\) and \(D''\) is the most general feature structure \(D\), such that \(D'\) \(\text{subsumes}\) \(D\) and \(D''\) \(\text{subsumes}\) \(D\). The partial ordering imposed by subsumption provides a framework or lattice for characterizing unification\(^3\). However, subsumption is only a partial order; thus, not every two feature structures have a subsumption relation with each other. The notion of combining the information from two feature structures, each of which may contain only partial information, to obtain a feature structure that includes all of the information of both, which still may be only a partially defined structure, is the central idea of unification. Conjunctive unification, referred to simply as unification unless an additional qualifier is stated, is an information combining procedure whose goal is to produce the most general feature structure that contains all the information from the unified feature structures with no additional information.

3.2 Feature Structure Representation: Enhancements for User Modeling

To allow ease of user model construction, we incorporated several enhancements to the basic feature structure: disjunctive features, range-valued and multi-valued features, and functional processing features. Each of these enhancements is discussed in turn in the following paragraphs.

Disjunctive features are distinguished by a specific set of values, at least one of which must be its true value. Disjunctive features are a logical construct that is useful in describing feature generalizations within the feature structure representation. Disjunctive features are of great utility in many areas where interpretation of user semantic meaning in context is important\(^5,10\). Disjunction leads to a more powerful and expressive feature structure representation in which to model user context goal information that is otherwise awkward to model using only conjunctive feature structures. Although disjunctive feature structures are of great utility, they are difficult to manipulate. When performing disjunctive unification, the system must make certain that all possibilities are maintained since the feature’s value need only be one of the values contained in the disjunction. The implementation of disjunctive unification involves “cross multiplying” the values of the two disjunctive features\(^3\).

The use of range-valued and multi-valued features is another extension to the basic feature structure representation. Features defined by range-valued and multi-valued terms may take on more than one value simultaneously. In our work, range-valued features are used primarily to allow for noisy sensor input data, uncertainty in the information derived from the sensor data and in the process of matching with the represented user model. Multi-valued features are utilized in feature structure rules as qualitative user goal and semantic information within the user model. Like disjunctive features, the extension of the feature structure representation to include range-valued and multi-valued features dictated a change in the basic unification procedure. However, unlike disjunctive features where only one of the values need be true, range-valued and multi-valued features require that all of the values of the feature be true. Due to this situation, the unification of two features with a multiple value is the most specific set of values that are true for both features, in other words, the intersection of the two sets.

The final enhancement, a functional processing type of feature structure, contains a feature value that is computed dynamically. The feature structure is allowed to have a functional processing value that, as a side-effect of unification, calculates a value for the feature using the indicated function. When the matching procedure encounters a functional processing value, matching is suspended while the value is calculated by the specified function contained in the feature value (the function to calculate the value may also be dynamically evaluated). The value returned by the function then replaces the functional processing value in the current feature structure if unification is successful. In this manner, many different
processing functions may be utilized dynamically by the CGA.

3.3 Rules as Feature Structures

Rules in a unification grammar define how to manipulate feature structures to build new feature structures. A key aspect of unification grammars is that a rule is represented as a feature structure. Application of a rule in the user model proceeds by attempting to unify the feature structure representation of the rule with the current feature structure representation of user model derived information. Rule application is permitted if the current feature structure does not violate any constraints represented as feature/value pairs of the rule. The unification process simultaneously enforces the constraints of the rule and builds the new feature structure that is specified by the rule.

3.4 Lattices of Feature Structures

Feature structures may be viewed as components within a mathematically defined bounded lattice, which allows for efficient CGA hypothesis generation and operator intent interpretation (utilized by the CGA for reasoning) and uses the basic operation of unification to drive the interpretation process. The feature structure representation of user models and their components within a partially ordered lattice enables the incorporation of additional ideas, such as abstraction and hierarchical representation, that allows the feature structures to be expressed at the appropriate level of detail and further enhances CGA flexibility.

4. User Modeling Techniques for CGA Behavioral Representation

Incorporation of user modeling for CGA behavioral modeling is occurring within system development for the Distributed Mission Training Integrated Threat Environment (DMTITE) project. Development for this project uses the open decision and knowledge architecture presented in the generic CGA architecture summarized in Section 2.1. Within the DMTITE CGA decision architecture, a user modeling methodology was most appropriate in the long-term decision engine of the CRC. The LTDE parallels the reasoning associated with operator goal and action processing for the employment of weapon platform assets. This type of information is readily expressible in user modeling terms and its representation using feature structures. In addition, the representation of a user model by feature structures, a declarative representation of knowledge, also provides benefits in the areas of representational flexibility, modifiability, and learnability. Therefore, the LTDE became the point for user modeling experimentation for CGA behavioral representation.

The goal of CGA decision making may be expressed within our framework of user modeling as a lattice of feature structures. The unification process produces a path through the enumerated lattice, generating a search lattice, to a possible goal state. An enumerated lattice is a lattice that could be produced given the entire set of feature structure models, feature structure rules, and feature structure inputs and consists of all feature structure unification paths that are theoretically possible. The search lattice is the lattice produced in the search through the enumerated lattice during the determination of user model state (CGA decision product) and contains only the feature structures and unification paths that are produced in the current search through the enumerated lattice to reach a goal.

The CGA long-term decision process, expressed as a path through the lattice of feature structures, represents a general search problem. Thus, CGA LTDE problem solving is represented as finding and applying a sequence of operators that will transform an initial system state, through a sequence of intermediate states, into a system goal state. Thus the LTDE product consists of finding and applying a sequence of feature structure rules that will transform the given world state and user model (initial system state), through a sequence of feature structures contained in the enumerated lattice that add derived feature information to the original input data by unification (intermediate states), into a system goal state containing the action of the long-term decision engine as a feature structure.

The critical problem solving step is to determine which grammar rules expressed as feature structures are appropriate at the given time and then to decide which features structures may be unified with the chosen grammar rule. At present, heuristics are used to control search through the lattice. These heuristics are implemented in the current system using meta-rules that reason about which grammar rules to unify. The meta-rule consequent is a grammar rule with which to unify. Once unification through the lattice is successful, the LTDE decision product is output to the arbitration engine. The AE incorporates CGA skill level and combat psychology modeling with the LTDE product to determine timing of CGA decision and action within the overall decision processing.

5. Conclusion and Future work

To address the lack of realistic behaviors for CGAs in DVEs, we adapted user modeling techniques for use within a generic CGA decision mechanism. The approach underlying user modeling, which is to represent action within a system so that the system can adapt to the needs of the user, can be readily adapted to the automated decision processing for CGAs. Our user modeling approach employs a unification-based grammar framework utilizing feature structures as the fundamental knowledge representation within the LTDE of our CGA. The incorporation of a user model within the LTDE is responsible for generating CGA behaviors that a traditional operator would usually perform for overall system control and utilization.

Our next step in CGA development will address additional knowledge acquisition for user model development. As coordinated behaviors provide a rich opportunity for overall system control and collaboration, the next major area for knowledge acquisition and implementation within the user modeling framework deals in CGA coordinated behaviors. Further testing is also necessary to determine the scope, level, and detail to
which the user should be modeled to provide the appropriate fidelity for CGA decision making.

6. References


