MacGyver Virtual Agents: Using Ontologies and Hierarchies for Resourceful Virtual Human Decision-Making (Extended Abstract)

John T. Balint
George Mason University
4400 University Drive, MSN 4A5
Fairfax, VA 22030
jbalint2@gmu.edu

Jan M. Allbeck
George Mason University
4400 University Drive, MSN 4A5
Fairfax, VA 22030
jallbeck@gmu.edu

ABSTRACT

Virtual Humans are becoming an ever more important asset in games, movies, and training simulations. The ability for virtual humans to select appropriate actions and objects for plausible behaviors is vital to creating believable and resourceful agents. For this purpose, a decision making methodology using hierarchies of ontologies for both actions and objects is presented. Objects are given semantic information such as affordances and physical properties. Affordance theory is applied to determine viable candidate objects for behaviors, and agents learn which objects are better choices than others.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence-Intelligent Agents

General Terms

Algorithms, Human Factors, Design

Keywords

Virtual character modeling and animation in games, education, training, and virtual environments; Modeling cognition and socio-cultural behavior

1. INTRODUCTION

Virtual Humans play vital roles in games, movies, and training simulations. These humans generally require extensive scripting in order to appear to reason about their abilities and surrounding environment. Smart objects, those tagged with semantic information, enable agents to act more intelligently and plausibly within their virtual world [5]. To do this, the agent needs to have an understanding of what objects are appropriate participants in each action, and which objects and properties of those objects are better suited for different situations.

Humans are resourceful creatures that are able to creatively use tools in new ways to solve problems given various contexts. For example, a person may use a table as a seat if no seats are available, even though tables are not specifically designed to be sat on. If virtual humans are meant to model real humans, they too should exhibit this kind of resourcefulness. Parameterized behaviors, such as those found in [1], allow for this flexibility as agents can create different action and object combinations given a rich virtual environment. However, certain combinations of actions and objects are more obvious than others. Using a hammer to drive a nail is commonsense, but in a pinch, a hard soled shoe could also be used. On the other hand, using a banana to drive a nail would not work. By finding a balance between what can go together and what should go together, virtual humans are able to create the appearance of being resourceful (though perhaps at a more common level than the famous MacGyver character).

A challenge to creating virtual humans that exist within a smart object environment (hereafter called an intelligent environment) is tagging the objects within that environment with important semantic information. Such information can create an ontology for each object. Some information would directly link an object to an action, such as affordances, while other information, such as physical properties, only contain object states. Organizing these smart objects into hierarchies creates a structured understanding of each object while allowing common ontological information to be stored at more abstract levels in the hierarchy. By attaching semantic information at the highest level of the object hierarchy, we do not need to store large amounts of similar information, and therefore can use our decision making formula in more complex environments. It has been shown that humans store information in a similar manner [2], and it has the added benefit of making searching though different options simpler by localizing the search to parts of the hierarchy.

To allow an agent to create interesting and plausible action object combinations, we propose a decision making formula to allow virtual agents to use pertinent semantic information within their environment. This system takes advantage of affordance theory [4] to find viable participant objects for an action, and then scores the set of useful objects using other semantic information and the agents own understanding of each object.

2. METHODOLOGY

2.1 Semantic information representation

Certain forms of semantic information link actions with different objects and their positions. Others [3] have used the term affordances for this form of information, and we choose to do the same. Affordances are semantic information attached to objects, and we attach them to different levels in the object hierarchy to signify classes of objects that can be used for an action.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013),* Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

Crucial to our object selection technique is the way we represent certain kinds of semantic information, such as physical properties and states of objects. Semantic properties, hereby denoted p, are categorized into sets, S, and these sets make up the set of semantic information. Objects may contain one property per set, and may be marked with many different sets. For example, properties p_i and p_j are members of set S_m, and property p_k is a member of set S_n. An object may have either property p_i or p_j, without an effect on its ability to also have p_k. There is generally overlap in the semantic information held by objects in a hierarchy of objects. Both forks and spoons, which are *silverware* objects, could be silver. Semantic information such as this can be remembered at higher levels in the hierarchy, removing repetition and therefore condensing the amount of data the agent must maintain in its world view. Physical properties and affordances are added to the system by a human user, as automatically determining semantic information is currently outside the scope of this project.

Semantic information can also be contained in the actions the agent performs. This information can either be data the action needs to successfully complete or properties the action changes. For example, a cleanse action would have a property indicating that the targeted object is *clean* at the end of the action. Note that this data should only update the world model if the action is successfully completed. In this case a property of the object would be changed from *dirty* to *clean*. These object properties can be referenced by an agent to indicate whether or not an object is a desirable participant in an action (e.g. it is more desirable for an agent to serve food on a *clean* dish).

2.2 Object Selection

When an agent is deciding an object's feasibility from a list of candidate objects, it first determines which objects afford the action in question. As the affordance ontology is evolved to attach concepts to the highest levels of the parent object, a small subset of high level objects are chosen though searching. We have the agent exhaustively search through the hierarchy of meta-objects, as the number of instanced objects is generally greater than the number of meta-objects. The set of instanced objects that the agent considers are compared to those meta-objects to determine if the higher objects are parents in the object hierarchy. Objects that are not children of the higher level objects are removed from further consideration.

$$\text{score}=e^{\text{learnt}+\text{ap}}+\frac{a}{b+\text{distance}}$$

Equation 1: The object scoring function. Learnt is the learnt parameter for the object-action combination, ap are action properties whose function is defined in equation 2, and distance is the Euclidean distance from the agent to the object. The parameters *a* and *b* are tunable constants.

Once the agent determines the function of an object, it scores the object based on the parameters in Equation 1. Agents learn combinations through simple reinforcement learning. An agent may select an object for an action, and receive interactive feedback on the combination. The agent stores this information much like affordance information, and is able to use this information in subsequent decisions through multiple scenarios.

$$ap=\sum_{a=0}^{n}F(O_i,p_a)$$

$$F(O_i,p_a)=\begin{cases} 1 & p_a \square O_i \\ -1 & !(p_a \square O_i) \text{ and } \forall p \square O_i \text{ s.t. } p,p_a \square S \\ 0 & \text{otherwise} \end{cases}$$

Equation 2: The formula for computing a score given action and object properties. O_i signifies the object in question, p_a is the action property in question, and S is a given property set.

The agent also determines a score for action properties, seen in Equation 2. As can be seen from Equation 1, the learnt parameter and action properties parameter are equal in their importance. Since action properties are attached to actions the agent performs, they can be considered prior knowledge on the needs of an action.

The distance from the object can be calculated either from the agent or another object already selected to be used in the action. For example, consider an agent that wishes to grab a drink and sit. Once the agent chooses a drink, it is understandable for the agent to choose a place close to that drink to sit. So, the agent would consider the distance from the previous object to the next object choice instead of the agent's position during the decision. This prevents the agent from backtracking.

3. Future Work

The described methodology enables agents to create and perform different action-object pairs in a semantic environment. In the future, we would like to allow the agent to be able to move semantic information of objects and actions to their most general position within a hierarchy. This will allows the agent to have a more generalized understanding of their virtual environment, and to create a more variable scenario without the need for scripting. We also believe that combining this methodology with a semantic database will give the agent a greater understanding of the world, and automatic tagging from such a database would ease the creation of rich virtual environments.

4. ACKNOWLEDGMENTS

Partial support for this effort is gratefully acknowledged from the U.S. Army SUBTLEMURI W911NF-07-1-0216. We also appreciate donations from Autodesk.

5. REFERENCES

[1] Bindiganavale, Rama, Schuler, William, Allbeck, Jan M, Badler, Norman I, Joshi, Aravind K, and Palmer, Martha. Dynamically Altering Agent Behaviors Using Natural Language Instructions. Autonomous Agents (2000), 293-300.

[2] Botvinick, Matthew M, Niv, Yael, and Barto, Andrew C. Hierarchically organized behavior and its neural foundations: A reinforcement Learning perspective. *Cognition*, 113, 3 (December 2009), 262-280.

[3] Donikian, Stephane and Paris, Sebastien. Towards Embodied and Situated Virtual Humans. In Motion in Games (Utrecht 2008)

[4] Gibson, James J. The Ecological Approach to Visual Perception. Psychology Press, 1986.

[5] Kallmann, Marcelo and Thalmann, Daniel. Modeling Objects for Interaction Tasks. (Lisbon 1998), Eurographics Workshop on Animation and Simulation, 76-86.