Automated Generation of Plausible Agent Object Interactions

Tim Balint and Jan M. Allbeck

Laboratory for Games and Intelligent Agents
http:\\cs.gmu.edu\~gaia
George Mason University, 4400 University Drive, MSN 4A5
Fairfax, VA 22030
(jbalint2, jallbeck)@gmu.edu

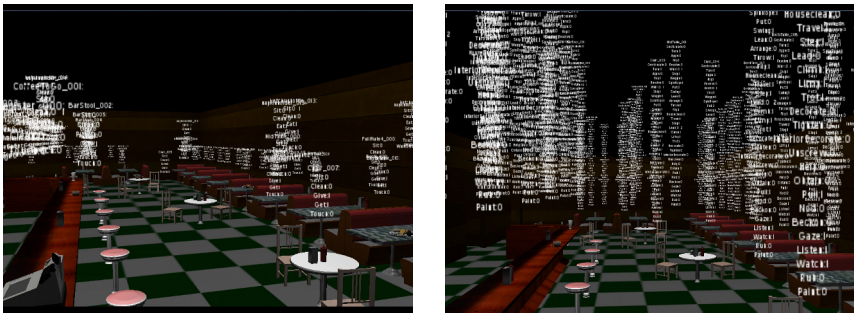
Abstract. To interact in a virtual environment, an agent must be aware of its available actions and the objects that can participate in these actions. Work such as Kallmann's SmartObjects allows for this information to be encoded by a simulation author, but these encodings tend to be constrained to individual graphical models. Furthermore, there may not be agreement on the proper operational information between simulation authors. To create consistent object operational information, we have devised a method that uses natural language lexical databases to parse and assemble this information. The method uses a two step process: 1. The names of motion clips are disambiguated using their name and a list of keywords; 2. Objects are connected to the actions by resolving the operational elements of a given verb. This method is tested on several common, publicly available action sets and the accuracy and coverage of our method are measured.

Keywords: Behavior Planning and Realization; Authoring/Reuse/Tools; Applications for Film, Animation, Art and Game

1 Introduction

Virtual humans play a vital role in games, movies, and training simulations. Some virtual humans can reason about their environment provided information critical to that reasoning is inherent in the objects of the scenario. One important type of semantic information for objects is operational information, which instructs virtual agents on the way in which an object is used in an action. However, for a large scale virtual environment containing several hundred objects, attaching semantic operational information to each object is a tedious and time consuming process (See Figure 1). Creating ontologies of objects allows similar objects to be grouped together into a forest of Directed Acyclic Graphs. Operational information can then exist at different levels of an ontology and propagate down to children, alleviating much of the burden a simulation author would face in creating a virtual environment [5].

One technique to encode semantic information into a virtual environment is to use SmartObjects [9]. This still requires a simulation author to encode all the connections, which for a large scale simulation with several different actions can also become tedious. Correct information can be automatically generated



(a) (b)

Fig. 1: Object operational data obtained (a) by previous manual methods for making object-action associations and (b) through the automated method presented in this paper.

through the use of Affordance Theory [6] if the information is purely visual. For functionality that is non-visual, other techniques must be developed. Managed online resources, such as WordNet [20] and FrameNet [1] contain academically generated action understandings and have been designed to interconnect. FrameNet in particular encodes different semantic roles in sentence structures. Its use would allow a system to distinguish the purpose of an object's participation in an action and the object's specific role, as seen by the different objects that can participate in an *Cook* action in Figure 2. As these are general use natural language dictionaries, the data contained in them needs refinement to be appropriate for use with virtual humans. The generality of lexical databases also makes it challenging to determine the proper meaning of an action, which is required to use the lexical databases.

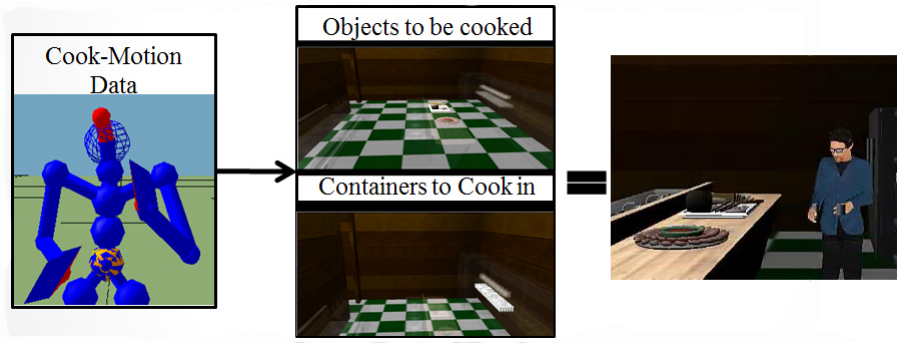


Fig. 2: A diagram showing how operational information is used by an agent.

In spite of these limitations, the scale and accessibility of natural language lexical databases make them a useful tool in the design and creation of object

operational information. To ease the work of a simulation author, we present an automated method for obtaining object operational information for use by virtual agents in large scale virtual environments. Specifically, the contributions of this work are:

- Automated generation of a hierarchy of actions from the names and short descriptions of motion clips.
- Automatic connection of the generated action hierarchy to an object ontology in a consistent and expandable manner.

These contributions are used to supply objects in a virtual environment with semantic information about their use given the motion clips available in the scenario.

2 Related Work

Cognitively, object operational information is dominated by Gibson's affordance theory [6]. This theory explains that physical humans can determine how an object is to be used by its physical properties. Affordance theory has been used by several virtual agent systems [3, 17, 14], and is a powerful technique in an agent decision making process. However, following the exact definition of affordances only allows an agent to know the usefulness of an object by its visual properties. Other properties, such as chemical and physical properties, are left out. A well used example of this is that it cannot be known just by examining an object whether or not can be safely consumed. Therefore, instead of using affordance theory for object operational information, we use natural language lexical databases to linguistically connect virtual objects to actions through the use of their names. This is analogous to a general domain knowledge approach for virtual agents.

While affordance theory has made strides in connecting virtual agent actions to semantic objects, other methods have tried to circumvent the visual constraint and ease the amount of necessary affordance information, even when using the term affordances. Pelkey and Allbeck [15] used natural language lexical databases to determine properties of an object, and Lugrin and Cavazza [12] uses semantic information to create non-kinematic visual effects in a game environment. Peters et al.[16] created a follow on technique to SmartObjects that specifically encoded operational information in the form of slots. This allows virtual objects to be used in several simulations actions, and were specifically designed for gaze behavior. To create operational information, these methods require explicitly written out connection between objects and actions, which our method does in an automated fashion. Instead of trying to generate affordances, Heckel and Youngblood [8] constrained the list of possible affordances based on situational awareness, but still requires affordances to be in the system.

There has been much work in showing the importance of object operational information, especially for use by virtual actors in a semantic environment. Much of this work is focused on ways to reason over the attachments as long as the attachments are already there, with some showing the usefulness of reusable operational information in simulations. Our work takes steps to fill in this knowledge gap by semi-automatically creating attachments from linguistic information about the objects and actions available in a scene.

3 Natural Language Lexical Databases

Converting action names into object operational information requires understanding the *sense* of a given action, that is, the proper context a given action would exist in. This can manifest itself through either its related words or meaning, as seen in Table 1. As another example, the phrase *pick up* has at least two senses, *to lift an object* or to *understand an idea*. The type of objects that can participate is vastly different depending on which sense the user desires, and different animations for a virtual human would accompany each sense. To disambiguate polysemous words, physical humans will examine a word's context, and determine which definition best fits the context. This is known as word sense disambiguation. A good survey of the topic can be found in [13]. Most word sense disambiguation techniques focus on a given context, such as the sentence in which the word is found. Unfortunately, even well named motion clips and processes do not readily appear in full sentences with enough context to determine their sense.

Term	**Synonyms**	**Definition**
Cook	n/a	Prepare a hot meal
Cook	Fix, Ready, Make, Prepare	Prepare for eating by applying heat
Cook	n/a	Transform and make suitable for consumption by heating
Cook	Fudge, Manipulate, Fake, Falsify, Wangle, Misrepresent	Tamper, with the purpose of deception

Table 1: An example of the polysemous word *Cook* taken from WordNet. Only the verb senses are shown.

For both word sense disambiguation and object operational connections, the choice of knowledge bases impacts our system's ability to attach operational information to objects. For our system, we use two knowledge bases that capture the linguistic understanding of object operational information, WordNet [20][1] and FrameNet [1][2]. FrameNet contains operational information in the form of **Frame Elements (FE)**, which are the participants of a frame of a verb. For example, a *sleep* verb contains the FEs *Sleeper* and *Place*. Shi and Mihalcea have connected FrameNet frames to WordNet senses [18]. We use WordNet to determine our initial sense due to its larger collection of verbs and the more prevalent parent-child relationships. The tree structure of WordNet allows verbs to be understood more generally, which we exploit in our process.

Figure 3 illustrates how these lexical databases fit into our overall system. The list of actions available to be performed in the scenario are disambiguated, mapped to WordNet verb synsets, and formed into an action ontology. The FEs of the verbs are linked to WordNet object synsets and ultimately to object models in the virtual world. This linkage defines how objects can be used (i.e.

[1] Also found online at http://wordnet.princeton.edu
[2] Also found online at https://framenet.icsi.berkeley.edu

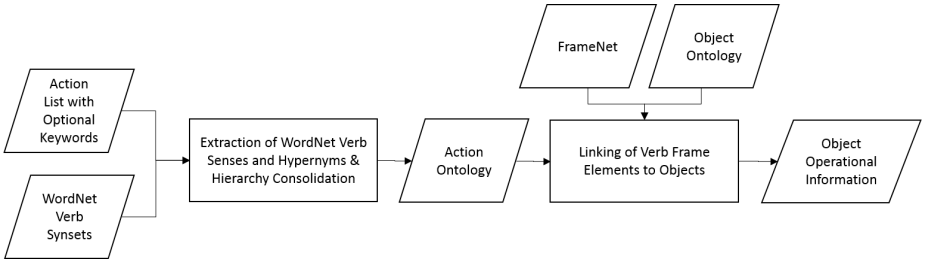


Fig. 3: System Overview

operated) in the actions. The following sections will provide more details about these system processes.

4 Action Ontology Creation

4.1 Determining the Sense of an Action

To determine object operational information, our method first reasons about the actions that a simulation author has created. These actions can come from a variety of sources such as low level atomic animations or procedural controllers (e.g. picking up an object). The sense candidates chosen from WordNet are directly related to the name a simulation author provides for the action. Therefore, the level of detail and ability of an author to describe an action has a large impact on the system's ability to derive its sense, with more general methods being used to disambiguate fuzzy situations as is done cognitively by physical humans [2]. To enable this ability in our system and provide more information to an agent reasoning about actions, we maintain much of the synset hierarchy as an action ontology. The hierarchical nature of verbs also lends itself naturally to a hierarchical approach, as seen in Figure 4. Our hierarchical method examines word senses using information on the word itself, its children, definition, and the relationship to other found senses.

Unlike previous word sense disambiguation methods, our method examines a word w given a small constrained set of user provided keywords $k = \{k_1, \ldots, k_n\}$ and determines the proper sense of w from the set of candidate senses $s = \{s_1, \ldots, s_m\}$. w can be any word or word phrase, such as *pick up* or *duck and shoot*. For the verb *cook*, a user might provide the keywords *food* and *heat*. As our domain specifically focuses on actions for virtual agents, we assume that w contains at least one verb that can be performed by a virtual actor. This is an important assumption as it greatly prunes the search space of candidate senses for a given verb. It is also a reasonable assumption for our target applications. The keywords should contain some context to the given sense of w, however, due to the descriptions found in animations, this may not always be the case. The name of the motion itself may only give a partial clue to its nature. We process w to determine s by first searching for s using w. If no results are found, we determine if w is a phrase, and search for s from each verb in the phrase. If no results are found at this stage, s is considered unresolvable.

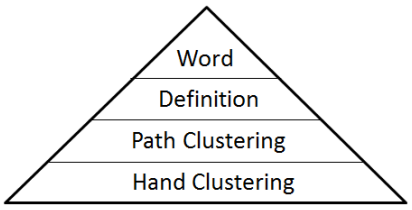


Fig. 4: The Multi-pass technique's testing conditions. Each level determines word senses with the most precise methods higher up in the pyramid. Techniques lower in the pyramid are less precise but have greater coverage.

Once our system has determined the set of candidate senses s, we search for the most likely candidate sense, s_{found}, by testing s against each method seen in Figure 4 and described in more detail in the following paragraphs, using Equation 1. α is used to reject s_{found} for low matching senses, and allows other methods to be tested in an attempt to find a better match, while not undermining high probability matches. Through testing we have found that $\alpha = 0.3$ provides a strong threshold while not being too discriminating.

$$s_{found} = argmax(method(s)) > \alpha \quad (1)$$

Word Disambiguation: The first methods used in our system attempts to disambiguate using only the sense set s and a set of k, similar to the disambiguation of agent commands done in [4]. As was done in [4], we use the lemmas of a sense, which are closely related to synonyms, as well as the parent verbs in WordNet's hierarchy, and determine the number of k that matches each candidate in s on these metrics. We have also added a third technique, which compares the lemmas of the sense's child verbs in WordNet's hierarchy to k by performing a Breadth First Search on the subtree of the sense. In order to get a percentage score to compare to α, we divide the total matches by $|k|$. For our example of *cook* with keywords *food* and *heat*, this method results in a score of zero. Neither keyword is found in the set of synonyms for *cook*.

Definition Disambiguation: If the above technique cannot disambiguate the sense of the word from its given components, we then examine the definition of each sense. The definition of a sense is a more relaxed search, and provides context to each sense in s. Testing the definition of a sense against k is a relaxed string matching approach. If a keyword matches any of the words in the definition, then it is considered a match, and is scored similarly to the word disambiguation technique. We also determine a percentage score for these techniques by dividing the total matching found by $|k|$. Here our *cook* example yields a score of 0.5, because the keyword *heat* is found in one of the sense definitions, but *food* is not. As we have found a sense whose score is higher than our α of 0.3, this sense is chosen.

Path Disambiguation: The final automated technique used to determine the sense of w is to compare s to the already disambiguated verb senses. This technique follows [11], which has been previously used to connect WordNet senses and FrameNet frames. Provided the system has determined one correct sense, this technique uses the Wu-Palmer Similarity [21] of s to determine the percent similarity to the already found senses of actions. This method is an iterative approach, and so will attempt to connect senses until no new senses are found.

Hand Clustering: When there is not enough information to disambiguate s or if none of the techniques are able to resolve the sense of a verb, then manual disambiguation is necessary. Our system provides tools for a user to choose the

correct sense given a definition and list of synonyms of each sense in s. If the user cannot find a given sense at this stage, then the action name is considered unresolvable and connected to a created action sense called *Human Action*, which does not have any object operational information.

4.2 Tree Generation

After the initial examination and disambiguation of verbs, we build an IsA action hierarchy, modeled from WordNet's verb hierarchy. This allows a virtual agent to reason about an action using a broader definition (e.g. being able to understand that a *pirouette* is a *turn* or that *baking* is a form of *cooking* which is in turn a *creation action*). To construct an action forest, we first generate the full parent list of each disambiguated verb (See Figure 5(b)). We then perform a node comparison, using Wu-Palmer path similarity [21] to the leaves of all previously generated trees to find candidate subtrees that the sense may belong to (See Figure 5(a)). We then examine all nodes in the list and candidate sub-trees, searching for a direct string match for a given verb sense. When one is found, that node and all child nodes are directly connected to the sub-tree. This allows a large amount of information to be reasoned over, increasing the total coverage of the system.

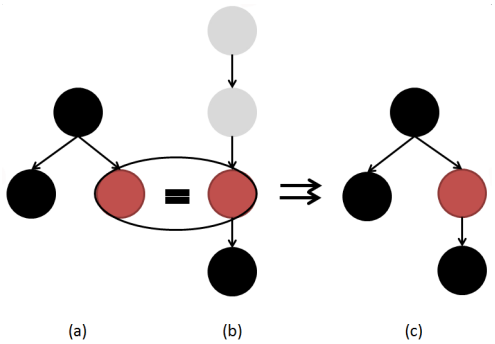


Fig. 5: A node in an action hierarchy being added (b) matches a node in an existing subtree (a). The hierarchies are merged at the common node and the ancestors of (b)'s matching node are removed (c).

Once the actions' word senses have been determined and the action ontology created, the actions can be linked to object participants, thus providing agents with information about how the objects can be used in virtual worlds.

5 Object Operational Information

Object operational information requires a connection between a given action and the objects that can be used in it. For several basic actions, this simply forms a triplet *(agent, action, object)*, where *agent* is the agent initiating the *action* on

an *object*. However, actions can become very complex, requiring not only knowledge of *what* objects can participate, but *how* they can as well. For example, an agent mopping a floor requires not only a space in which to mop, but an instrument with which to perform the action. This creates a two-fold process, in which the correct sense of a frame must be disambiguated. Disambiguating the correct ordering of such an action combination is essential to streamlining an agent's decision making process [3].

After the sense of each action has been determined by the system, there is enough information to determine and link the set of actions to a set of physical objects found in a scenario. Figure 6 shows an overview of the connection step. Each Frame Element (FE) is matched against an ontology that contains graphical objects from one or more virtual environments and an understanding of their generalization. For the example of *cook*, one of the function elements is *container* (See Table 2). If the simulation contains a graphical representation of a *bowl*, then the ontology should reflect that a *bowl* is a type of *container*, which is ultimately a *physical object*. The connection between *cook* and *container* would then be reflected in the operational information, allowing the system to use a *bowl* for *cooking*. A similar connection is made between *food* and *cook*.

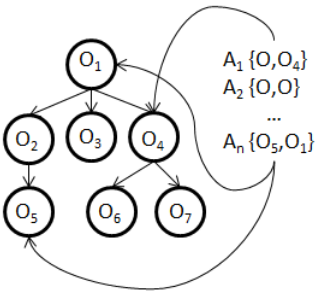


Fig. 6: A pictorial overview of the connection step. The object parameters of actions are linked to object types in the object ontology.

The generation of an object ontology is outside the scope of this work. A method for automated object ontology construction can be found in [15]. If a FE cannot be matched to the ontology, it is rejected. As we are only dealing with graphical semantic objects, we restrict the objects considered to a physical object ontology. This specifically removes two types of FEs: information that cannot be visualized by graphical models, such as *Purpose*, and ones that are simply not available to the simulation author, such as if they do not have a *heating instrument* for the *cook* action. While important to a full representation of an action, non-physical nouns that are described in an action (such as an action containing a *manner* object) are outside the scope of this paper.

We connect objects to found sense names using techniques identical to the string matching, partial string matching, and Breadth First Search methods used in the previous section and in [4] on FrameNet FEs. Some FEs contain a semantic type, which is a generalized explanation of that FE and can contain parent and children semantic types. Unlike word disambiguation, we want the FE to be as specific as possible, and therefore only search on the children of the semantic types.

Creating object operational information in this manner provides an upper bound on the types of objects that are connected to that action. Agents can then use this upper bound to help determine possible candidates by examining the object ontology and their environment. The utility of our method is therefore

Element	**Object Type Connection**
Cook	Semantic type linked to Agent
Produced Food	Food provides link to Food object type
Container	Links to Container object type
Degree	*Not a physical object*
Heating Instrument	*While the sample environment includes a grill, the object hierarchy does not recognize it as a heating instrument.*
...	...

Table 2: Part of the Frame Elements for the verb *Cook*. Elements not picked up by the system are in italics.

dependent on the ability to disambiguate the sense of an action's name and connect important components of the action to objects.

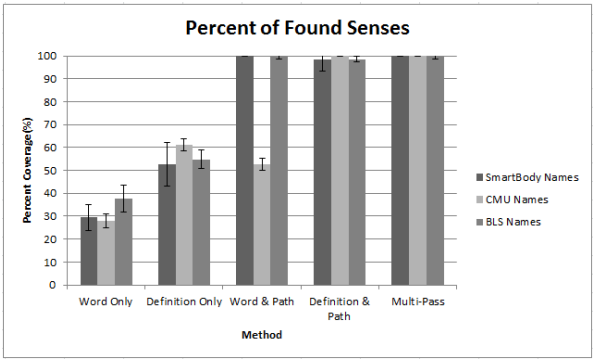
6 Analysis and Results

In order to test the ability of our method to determine the sense of an action from its name and keywords and provide object operational data to virtual environments, we have formulated and tested several hypotheses:

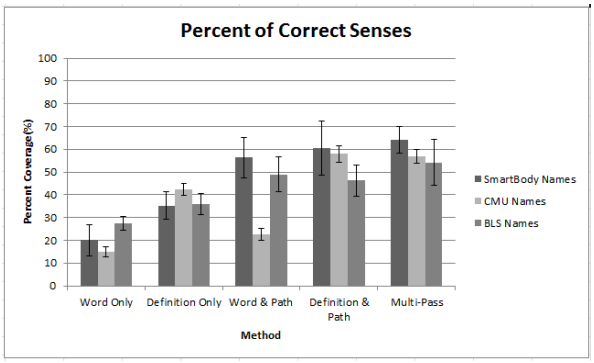
- **H1**: A consistent, extensible action ontology can be created using well named motion data and online lexical databases.
- **H2**: Using a mutli-pass method is a more accurate way of determining the sense of a verb obtained from an animation's name.
- **H3**: The resolution provided by FrameNet and a virtual object ontology covers the usable objects in a scenario.
- **H4**: An action ontology derived from WordNet provides more extensive and accurate coverage than using FrameNet alone.

To test **H1** and **H2** we used action names from CMU's motion capture library [7], a scan of the SmartBody documentation [19], and a higher level data set that represents common behaviors listed in tables provided by the Bureau of Labor Statistics (BLS) [10]. There is some overlap between senses from the SmartBody and CMU action sets. This creates a list of sixty actions, fifteen actions, and forty two actions respectively that a virtual human would be capable of. Naturally, behaviors in the BLS set do not inherently correspond to virtual human capabilities, but the set provides us with another source of potential behaviors. We create a ground truth for each data-set that contains the correct sense for each action by hand examining the senses in WordNet. For each dataset, we also create two list of keywords: a definition list from the Merriam Webster's dictionary definition (generally the longest words in the definition) and a list containing synonyms from Merriam Webster's dictionary[3]. We then create several tests from the set of keywords, choosing either the definition or synonym set for each action. These are used to compare our overall method to each of its components. The results can be seen in Figure 7. As the path similarity metric requires one action sense to examine WordNet paths, we combine that method with both our word metrics and definition metrics.

[3] The definition and synonym lists can be seen in the additional materials.



(a)



(b)

Fig. 7: The percent of found (a) and correct (b) senses for our ten sample test. Error bars represent one standard deviation. A single factor ANOVA analysis provided a negligible p-value for both figures, with a Tukey-Kramer test showing significant difference between our method and the word only, definition only, and word with path methods for CMU's data set, and word only and definition only for Smartbody's data set. A Tukey-Kramer test shows significant differences between all methods and the multi-path method for the BLS data set.

As can be seen from Figure 7a almost all of the multi-pass methods that included [11] were able to overwhelmingly find a sense to attach to an action's name. This means that, more often then not, using [11] will allow for some sense to be found and an action ontology to be generated for most of the actions without the need for an author to manually add them. Using our word disambiguation with path disambiguation for CMU's data set was an exception. This is because this method did not find enough senses using word disambiguation to connect senses using [11]. A lower α value would allow the process to find more senses, but may reduce the overall accuracy of the system. As a result, the high percentage of found senses confirms **H1**. When examining Figure 7b, there is an overall decrease in the system's ability to choose the correct sense . Therefore, while [11] allows for a hierarchy to be created, it may not be the correct one.

However, compared to methods that did not use a path method, Figure 7b shows that a multiple pass method is preferred when creating an action hierarchy from action names, confirming **H2**.

We performed an additional experiment to test hypothesis **H3**, by determining the percent coverage for both methods using a generated object hierarchy with over 100 leaf objects constructed based on the work of [15][4]. Ground truth for object operational information is obtained by using the corresponding action hierarchy from the ground truths used in the analysis of **H1** and **H2**, and participant connections are chosen even if there is not a corresponding connection in the object tree. Our method then computes coverage for each of the found object trees, using our automated method. We also compute a maximum ground truth using the ground truth action hierarchy for each data set. At this stage, we also tested **H4** by using the found action senses with and without the object hierarchy. The results can be found in Figure 8.

When using an action hierarchy to get a fuller definition of an action, it can be seen from Figure 8a that the closer to ground truth the starting senses are, the closer the expected connections are to the maximum coverage using our method. As there is a statistically significant difference for each dataset using each method, we cannot confirm **H3** and the true ability of the coverage. We can infer from the operational connections in the ground truth seen in Figure 8a that our method can resolve over half of the operational connections between our object and action ontologies. This already greatly reduces the effort needed by a simulation author and encourages further work in this area.

When no hierarchy is used, the ability to determine expected operational information decreases drastically, as seen by the low percentages in Figure 8b. This should be expected as the likelihood for a WordNet sense to have a corresponding FrameNet frame using [18] will increase when a more general sense understanding can be exploited. This confirms **H4**.

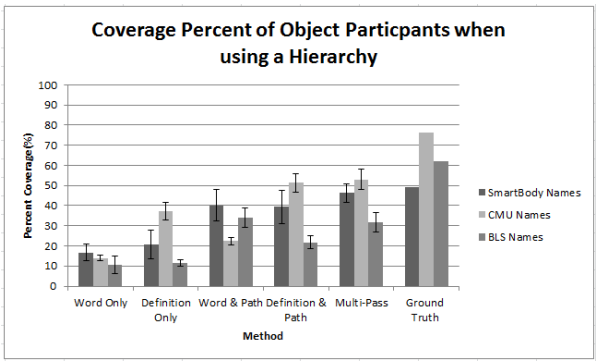
6.1 Connections with Virtual Humans

Determining the operational information of objects and actions in a scene can provide knowledge as to the set-up of a command to a virtual human. However, it is important that the motion a virtual human is performing is compatible with the objects that have been assigned to the action. To demonstrate this, we use more general behaviors from our BLS data set with sub-actions being generated by motion clips from CMU's motion library. The motions are performed using a standard SmartBody skeleton [19] and the animations from the motion library.

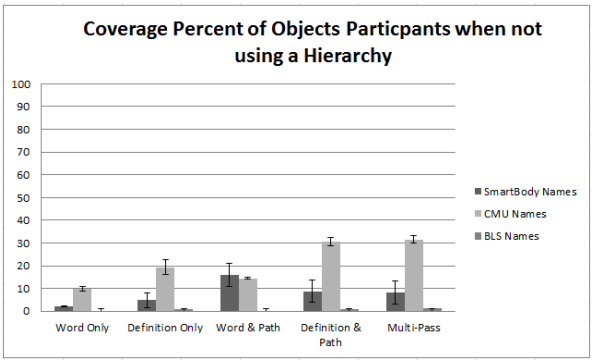
We place our virtual human in a diner that contains several hundred objects normally found in such an environment. We then instruct our agent to *Cook*. The results can be seen in Figure 2 [5]. It should be noted from this that the agent examines its environment and chooses objects that are appropriate to participate in the *Cook* action. This is using only the found connections, and not the final ground truth operational information not found in the object ontology. The found objects are then passed to a *Walk* and *PickUp* action through a planning language, and must have connected operational information for each of those actions as well.

[4] The list of leaf objects can be seen in the additional materials.

[5] The full example along with others can be found in the accompanying video.



(a)



(b)

Fig. 8: The percent of matched object operational information compared to a ground truth assessment (a) when examining the whole action hierarchy and (b) using only the found senses of the actions. A two factor ANOVA between both graphs found a statistically significant difference with a p_value of 0.00194, and the difference between each data set to be statistically significant with a p_value of 0.007.

7 Conclusions, Limitations, and Future Work

In order to determine object participants for parametrizable actions, we developed a multi-pass word sense disambiguation technique and graph search methodology. These techniques require well defined motion data names and a hierarchy of possible object names for a given set of scenarios. Using multiple passes to determine senses ultimately leads to more object operational information. The resolved connections provide a first step to an automated approach of agent object interaction.

Our work depends on a simulation author correctly naming each of their motions, so that our system can reason about the names of our actions. The similarity between the correct output in Figure 7b and Figure 8a may show that a direct correlation exists between the system's ability to reason about actions

and its ability to connect actions to plausible object participants. Methods that are less variable in our sense understanding stage may allow for better coverage of operational information. Future work will explore this connection. Examining the total connections the system could make to the ontology also show a need to use more information than only objects to increase the coverage. This can also be explored by looking at operational connection methods that are not sensitive to the name of the object in the ontology or its corresponding FE. One possibly promising approach is to use properties, for example *ingestible* to supplement these connections. Future work will also explore their use for this problem domain.

Our current system examines each individual action and does not consider objects that might be needed for parallel combinations of actions. However, it is common for actions to be combined together, such as the two actions *sit* and *eat*. Future work will examine determining participants for combinations of actions and will include determination of possible and impossible action combinations. A related task that is also considered future work will examine automated determination of prerequisites for actions.

8 Acknowledgments

This work was partially supported by a grant from the U.S. Army Night Vision and Electronic Sensor Directorate (W15P7T-06-D-E402) and software licenses from Autodesk. We would also like to acknowledge Shib Duman for the creation of 3D models used in our examples and John Mooney and Jessica Randall for their contributions in refining the presentation of this research.

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: Proceedings of the 17th International Conference on Computational Linguistics - Volume 1. pp. 86–90. COLING '98, Association for Computational Linguistics, Stroudsburg, PA, USA (1998), http://dx.doi.org/10.3115/980451.980860
2. Botvinick, M.M., Niv, Y., Barto, A.C.: Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. Cognition abs/1109.2130 (2008)
3. Donikian, S., Paris, S.: Towards embodied and situated virtual humans. In: Egges, A., Kamphuis, A., Overmars, M. (eds.) Motion in Games, Lecture Notes in Computer Science, vol. 5277, pp. 51–62. Springer Berlin Heidelberg (2008)
4. Fernández, C., Baiget, P., Roca, F.X., Gonzílez, J.: Augmenting video surveillance footage with virtual agents for incremental event evaluation. Pattern Recogn. Lett. 32(6), 878–889 (Apr 2011), http://dx.doi.org/10.1016/j.patrec.2010.09.027
5. Flotyski, J., Walczak, K.: Conceptual knowledge-based modeling of interactive 3d content. The Visual Computer pp. 1–20 (2014)
6. Gibson, J.: Perceiving, Acting and Knowing, chap. The Theory of Affordances. Lawrence Erlbaum (1977)
7. Guerra-Filho, G., Biswas, A.: The human motion database: A cognitive and parametric sampling of human motion. Image and Vision Computing pp. 251–261 (2012)

8. Heckel, F., Youngblood, G.: Contextual affordances for intelligent virtual characters. In: Vilhjlmsson, H., Kopp, S., Marsella, S., Thrisson, K. (eds.) Intelligent Virtual Agents, Lecture Notes in Computer Science, vol. 6895, pp. 202–208. Springer Berlin Heidelberg (2011)
9. Kallmann, M., Thalmann, D.: Direct 3d interaction with smart objects. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology. pp. 124–130. VRST '99, ACM, New York, NY, USA (1999), http://doi.acm.org/10.1145/323663.323683
10. of Labor Statistics, B.: American time use survey. Tech. rep. (2010), http://www.bls.gov/news.release/atus.nr0.htm
11. Laparra, E., Rigau, G.: Integrating wordnet and framenet using a knowledge-based word sense disambiguation algorithm. In: Proceedings of the International Conference RANLP-2009. pp. 208–213. Association for Computational Linguistics, Borovets, Bulgaria (September 2009), http://www.aclweb.org/anthology/R09-1039
12. Lugrin, J.L., Cavazza, M.: Making sense of virtual environments: Action representation, grounding and common sense. In: Proceedings of the 12th International Conference on Intelligent User Interfaces. pp. 225–234. IUI '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1216295.1216336
13. Navigli, R.: Word sense disambiguation: A survey. ACM Comput. Surv. 41(2), 10:1–10:69 (Feb 2009)
14. van Oijen, J., Dignum, F.: Scalable perception for bdi-agents embodied in virtual environments. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on. vol. 2, pp. 46–53 (Aug 2011)
15. Pelkey, C., Allbeck, J.M.: Populating virtual semantic environments. Computer Animation and Virtual Worlds 24(3), 405–414 (May 2014)
16. Peters, C., Dobbyn, S., MacNamee, B., O'Sullivan, C.: Smart objects for attentive agents. In: Proceedings of 11th International Conference in Central Europe on Computer Graphics (2003)
17. Sequeira, P., Vala, M., Paiva, A.: What can i do with this?: Finding possible interactions between characters and objects. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 5:1–5:7. AAMAS '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1329125.1329132
18. Shi, L., Mihalcea, R.: Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 3406, pp. 100–111. Springer Berlin Heidelberg (2005)
19. Thiebaux, M., Marsella, S., Marshall, A.N., Kallmann, M.: Smartbody: Behavior realization for embodied conversational agents. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1. pp. 151–158. AAMAS '08, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2008), http://dl.acm.org/citation.cfm?id=1402383.1402409
20. University, P.: About wordnet. Tech. rep., Princeton University (2010), http://wordnet.princeton.edu
21. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics. pp. 133–138. ACL '94, Association for Computational Linguistics, Stroudsburg, PA, USA (1994), http://dx.doi.org/10.3115/981732.981751