Speaking with your Sidekick: Understanding Situated Speech in Computer Role Playing Games

Peter Gorniak and **Deb Roy**
Cognitive Machines Group
MIT Media Laboratory
20 Ames St.
Cambridge, MA, 02142
{pgorniak,dkroy}@media.mit.edu

Abstract

Speech and natural language are natural and convenient ways to interact with artificial characters. Current use of language in games, however, is limited to menu systems and interplayer communication. To achieve smooth linguistic communication with synthetic agents, research should focus on how language connects to the situation in which it occurs. Taking account of the physical scene (where is the speaker located, what is around her, when does she speak?) as well as the functional aspects of the situation (why did he choose to speak? What are his likely plans?) can disambiguate the linguistic signal in form and content. We present a game environment to collect time synchronized speech and action streams, to visualize these data and to annotate them at different stages of processing. We further sketch a framework for situated speech understanding on such data, taking into account aspects of the physical situation as well as the plans players follow. Our results show that this combination of influences achieves remarkable improvements over the individual situation models despite the very noisy and spontaneous nature of the speech involved. This work provides a basis for developing characters that use situated natural spoken language to communicate meaningfully with human players.

Introduction

Many of the latest multiplayer games provide ways for players to communicate with each other via natural language, either by using typed messages, or increasingly by speech. Given the increasing popularity of online multiplayer environments and their need for convenient and efficient ways to coordinate strategies, share information and socially banter, it should come as no surprise that game designers are resorting to language as a natural communication medium. The same need to coordinate, share information and banter exists between human players and synthetic characters in the game. In this case, however, current games resort to a combination of point-and-click interfaces and menu driven dialogues, often to the frustration of both game designers and players. These interfaces are awkward, unnatural and inconvenient. Using natural language and speech to communicate with in-game characters, on the other hand, seems like an overly hard problem due to the noisy and spontaneous nature of the language used and its syntactic and semantic complexity.

In contrast, we believe that natural language and speech can be used as a communication medium with synthetic characters if we leverage the same contextual information that human beings use - the physical, referential context and the functional, intentional context. We argue here that it is possible to capture many aspects of context because of the relative ease of forming a model of the situation the speaker finds him- or herself in when playing a game. In-game objects and characters are easily accessible in terms of their location, properties and visual appearance, and can serve as a basis for reference during speech understanding. On the intentional level, the design of the game provides many clues as to the players' possible plans and needs. Games thus provide an ideal research platform for studying how to leverage the situated nature of speech to produce better understanding algorithms in realistic conditions. At the same time, such research can feed directly into game platform design and lead to synthetic characters that reliably understand a player's speech.

In this paper, we first present a game environment based on Bioware's Neverwinter Nights (Bioware Inc. 2002) role playing game that lets us capture players' action and speech in a sample scenario. Acknowledging the amount and complexity of data captured in this way, tools to visualize and annotate them play a central role in the next section of the paper. We then demonstrate a simple version of a situational model for the game, one that captures aspects of the physical in-game situation and recognizes the player's plans. Our Framework for Understanding Situated Speech (FUSS) occupies the remainder of the paper, integrating the situational model into a probabilistic speech understanding process. We show that this approach, despite the present simplicity of the situational model, can disambiguate a significant number of referents using both the physical and the intentional parts of the model. Either part alone does not nearly perform as well. Finally, we sketch the use of such an understanding framework in building a prototype synthetic character that understands noisy, spontaneous speech commands in the game.

Related Work

Some work that involves understanding language using situation models includes Schuler's reference to a symbolically

encoded situation during speech parsing (Schuler 2003), Narayanan's interpretation of news stories using an action representation (Narayanan 1997), and our own grounding of spatial language in visual scenes (Gorniak & Roy 2004). In contrast to these approaches, which deal dealing solely with visual scenes or abstract action models, we here particularly focus on language that occurs when speakers share a common history and common future plans, making reliance on shared intentions a common occurrence. Of these related works, only Schuler also uses a speech recognizer as input (as opposed to transcribed speech or text), but does not maintain ambiguities all the way down to semantic interpretation as we do here. Speech sausages and probabilistic Earley parsing are well known in the speech recognition literature (Mangu, Brill, & Stolcke 1999; Stolcke 1995), and stochastic context free grammars have recently been proposed for plan recognition in other domains (Bobick & Ivanov 1998; Pynadath & Wellman 2000).

The Game

Figure 1: The in-game perspective of a player in Neverwinter Nights, playing the module used in this paper.

Neverwinter Nights (Bioware Inc. 2002) includes an editor allowing the creation of custom game worlds and has a large and active online player base. A typical in-game view as seen by a player of our game module is shown in Figure 1. The two player module is structured around a puzzle. To simplify dialogue aspects of the data, we only allow one of the players to speak. The other (played by the experimenter), is in the same real-world room as the first player (the study participant), but does not speak and does not act autonomously - he or she only does as instructed. In this way we restrict interaction to be similar to what commanding an intelligent but passive machine controlled character would be like. However, we do not restrict the language used in any way (except indirectly through the choice of puzzle), and the speaking study participant knows that a human being is listening to his or her commands.

Figure 2 shows the map of the puzzle used for data collection. Both players' avatars start in the large room in the

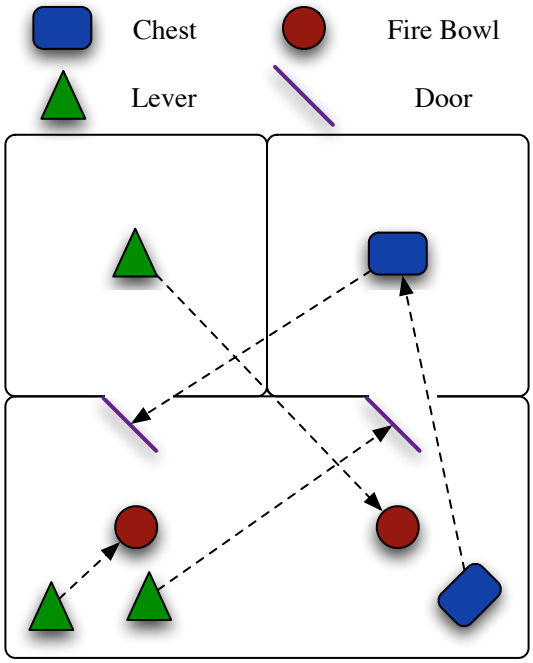

Figure 2: A diagram of the map used for data collection, with dashed lines indicating dependencies between objects.

bottom half of the map. The coloured symbols in the map represent objects (explained in the map legend), whereas the dashed arrows indicate the dependencies between objects that must be followed to solve the puzzle. The overall goal is to light both fire bowls at the same time. The players were only told about this overall goal, without knowing how to accomplish it. One chest contains a key that unlocks the second chest, which in turn contains a key that unlocks one of the doors. One of the levers opens the door to the second chest, whereas the other two levers (one behind the second door) light a fire bowl each. The puzzle cannot be solved by a single player due to timing constraints: the right door on the map can be opened with one of the levers, but it closes again after a short time, making it impossible for the same person to pull the lever and run through the door. Similarly, each fire bowl extinguishes itself after a few seconds unless both are lit, making it impossible for a single person to light both quickly enough. Participants usually solved the puzzle within 15 minutes.

During data collection, we recorded player's in-game actions, and his or her speech using a head-worn microphone. This yields a complete transcript of in-game actions and time-synchronized audio events. We ran our own utterance segmenter on the recorded audio, which produced 554 speech segments across 6 sessions (Yoshida 2002), and manually transcribed these utterances.

Visualizing and Annotating the Data

Figure 3 shows the interface we have developed for browsing and annotating the data we collected. At the bottom of

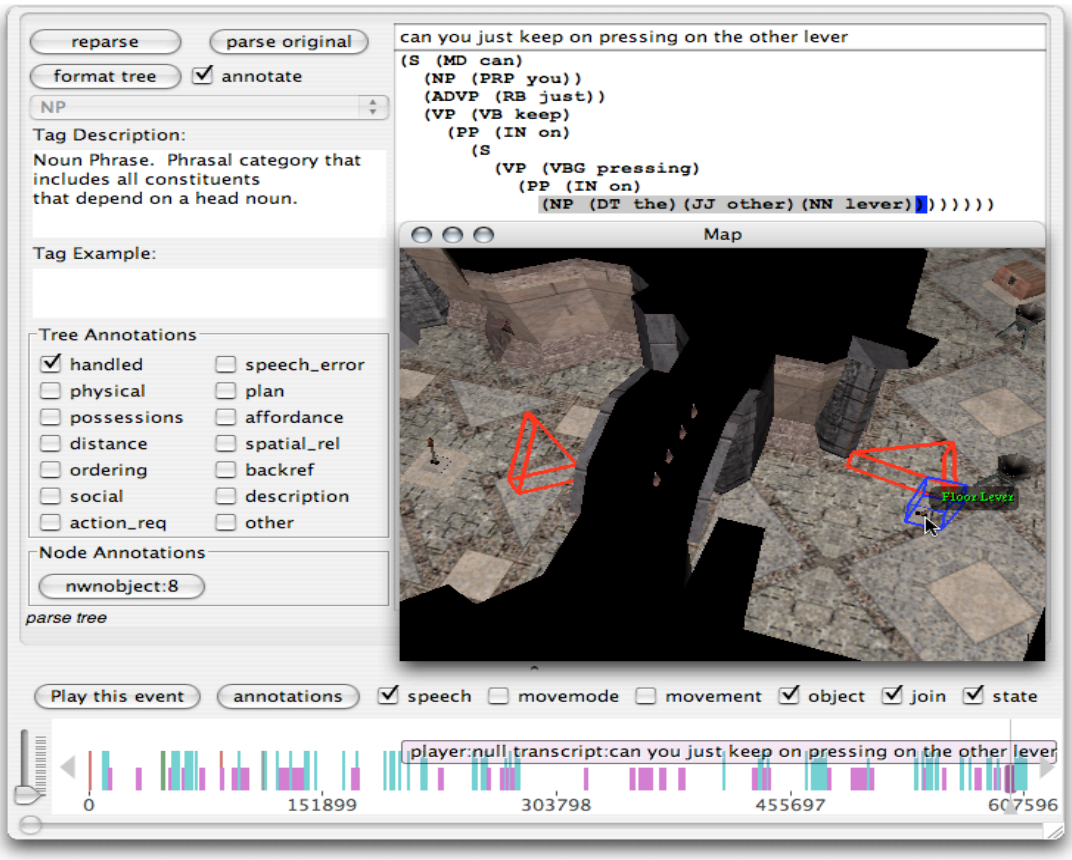

Figure 3: The tool used to replay, parse and annotated the utterances in their situational context

the window we find a panel showing a timeline of the events that occurred during the session. Events can be filtered using the check boxes above, quickly examined via overlayed information, and the user can zoom into and pan across different sections of the timeline. Furthermore, the timeline provides controls to step through a replay of the events, during which audio events are replayed and the map reflects player actions. The map panel is shown above and to the right of the timeline, and shows a picture very close to the one players see during game play, except that the camera can be arbitrarily controlled by the user and players are represented by red arrows.

The remainder of the window is filled with controls that let the user annotate a specific utterance. In the figure, the utterance "can you just keep on pressing on the other lever" has been selected. Above the map is a parse tree of the utterance. We initialize parse trees with the Stanford Parser Stanford Parser (Klein & Manning 2003) using a standard grammar for written English. This does not capture many of the phenomena encountered in spontaneous, situated speech, and so the parse tree panel allows the user to correct the parse tree, which is interactively re-formatted. The controls on the left show information about the currently selected syntactic node, and allow for re-parsing of the original utterance. Below these controls are the annotation markers for the current utterance. While the top set of these is largely unused in the study presented here (except to exclude noisy or off-topic utterances), the node annotation button below lets the user select a referent for the currently highlighted syntactic constituent. In the case shown, the user has selected the noun phrase "the other lever" and used the map panel to indicate the lever this utterance refers to, which is translated into the appropriate reference indicator by the interface.

Using this tool, we selected 90 utterances that contain noun phrases directly referring to a physical object in the game world, such as "activate the lever for me" or "can you come over here and pick this", but not "do that again" or "on the left". We annotated each of these noun phrases with its correct referent. We built a closed vocabulary trigram language model for the speech recognizer using the transcripts from the other sessions. The speech recognizer produced utterances of an average length of 23 words, while the transcribed utterances only average to 7 words each. Most of the extra hypothesized word slots stem from silences and noise within or around the actual speech utterance.

The Intentional Situation Model

While the physical situation model we employ for speech understanding in the game world simply consists of the physical objects present in the puzzle, the intentional model

is more complex. Despite the task players engaged in being an exploratory one, it was one with a clear goal (lighting both fires) and a limited number of ways to achieve this goal. We only show results about the predictive aspect of plan recognition in this paper, but it is important to keep in mind that players explicitly and implicitly refer to different levels within their own plan. For example "pull the lever" and "let me out" may be asking the other player to engage in exactly the same action, but bind to different levels of a plan hierarchy.

To recognize such hierarchical plans from players' actions, we employ a predictive probabilistic context free grammar parser, namely an Earley parser (Earley 1970; Stolcke 1995). Due to the predictive nature of the Earley parser it is possible to estimate the probability of a symbol being parsed at the next step by summing the probabilities of all rules currently being considered by the parser that have the symbol in question as the next symbol in their tail. During plan recognition, this lets us predict which objects the player will likely want the other character to interact with next, namely those that are involved in actions estimated as likely in the next steps of the plans currently in progress.

To train the plan parser, we abstracted the event traces of each data collection session into a higher level description that only contains the crucial events, such as object interactions and room changes. Subsequently, we hand-crafted a grammar that captures the sequence of events necessary to solve the puzzle in a hierarchical fashion, including multiple ways to solve the puzzle (e.g. opening a door to let the other character into a room vs. asking him to open the door). The grammar also includes sets of rules that have NOOP (a 'skip' symbol) as a top-level symbol so that exploration by the player is captured. We then estimated probabilities for this grammar using rule counts from the sessions other than the one being tested.

Grounding Language in the Situation Model

The predictions made by the plan parser now need be integrated with the words in an utterance and their referential targets to determine a most likely referent. We once more employ a probabilistic context free grammar parser, this time one that parses in a non-predictive bottom up mode, to robustly find grammatical fragments in the utterance. For each of the data collection sessions, we used the corrected parse trees of five sessions to learn a probabilistic context free grammar for the remaining one.

We augment the lexicon with information indicating the possible referents of words. As the parser completes syntactic constituents, it assigns a combination of their child constituents' referents to the newly created one. Methods like this one that drive semantic interpretation via syntactic parsing can achieve good referent resolution performance when they use more sophisticated physical situation models, as long as the situation is constrained such that the speaker's intention is clear from his or her utterance and the physical scene (Gorniak & Roy 2004). In our task here, a more sophisticated physical model that includes distance measures, containment relationships and object properties would enhance the performance when combined with semantic bindings that make use of these features. However, due to the strong planning aspect in the task used here, speakers did not use language containing explicit spatial references often. Rather, they most often referred to objects via simple deterministic noun phrases ("the lever"), despite there being multiple possible referents for such a phrase. Thus, the combination of physical and intentional model proposed here is not only beneficial, but necessary for referent resolution in this case.

The language parser yields a set of grammatical fragments with associated probabilities $P(w_{i...k}|G)$, the probability of words i through k in the utterance given the grammar G. When binding to possible referents as described above, the parser also produces $P(R|w_{i...k})$, the probability of referents given a segment of the utterance. Using Bayes' law we can convert this into $P(w_{k...i}|R)$ (the necessary prior will be discussed below), multiply it by $P(w_{i...k}|G)$ (using some believable independence assumptions) and apply Bayes once more to yield $P(R|w_{k...i}, G)$, the probability that the utterance fragment refers to an entity in the situation model. Note that while in the discussion here entities are assumed to be physical objects, they could equally well be other things, such as plan fragments produced by the parser, as discussed above. The necessary prior for this second application of Bayes' law is $P(R)$, the probability of referents. To integrate the intentional model into the understanding process, we use its predictions as priors at this point. In this way, bottom up physical language grounding and top-down intentional language grounding produce a coherently integrated estimate of the most likely referent.

Capturing Ambiguity in Speech

Typed text is one possibility for communicating with synthetic characters using natural language, but it is an inconvenient one in the game context. Often the player is using his or her hands for other game controls, such as commanding his or her own avatar, and typing messages in addition quickly becomes an annoyance. Speech offers itself as a convenient alternative, but as it is produced spontaneously in the middle of a game it is likely to be acoustically, lexically and syntactically noisy. Our data bears this out, and the Sphinx 4 speech recognizer we use for our system achieves only a 50% word error rate. This is partially due to the very small size of our sample data set producing the language models, but probably represents a realistic figure for a more complex game requiring a larger vocabulary as well as dialogue.

To overcome this problem, we have augmented the speech recognizer with 'sausage' generation facilities. Sausages are compact representations of possible hypotheses produced by a speech recognizer (Mangu, Brill, & Stolcke 1999; Hakkani-Tur & Riccardi 2003). Figure 4 shows a sausage from the data for the spoken utterance "Can you open the gate again." Nodes are shown in order of decreasing probability from top to bottom with the correct nodes highlighted. "<noop>" and "<sil>" are special words that stand for a possible word skip (i.e. the possibility that no word occurred in this slot) and a silence word, respectively. The example shows that the correct word is often not the one

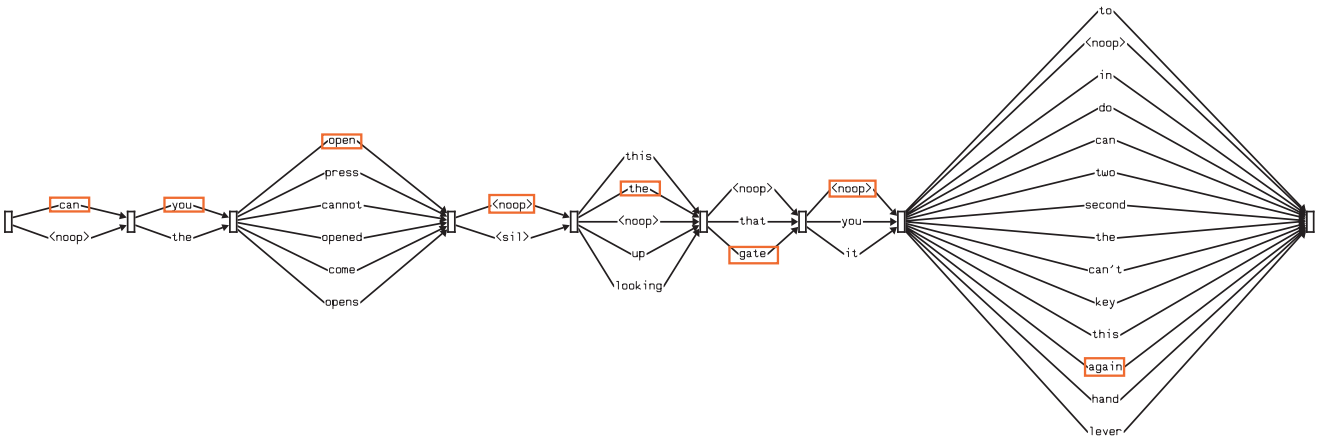

Figure 4: A sample sausage produced by Sphinx 4 for the utterance "Can you open the gate again".

with the highest probability, and that confusion varies from a single word choice to more than 10 choices. When following the path through the sausage that produces the fewest word errors (the "sausage oracle"), we achieve a word error rate of 23%. As we care less about transcribing the correct words and more about understanding the speaker's intention, this rate represents a more realistic speech recognizer performance for our task.

The sausage representation allows us to compute the likelihood of any path or set of paths through the sausage. Instead of single words, we offer each word in a slot with its associated probability to the probabilistic parser. The parser thus considers all possible paths through the sausage as is proceeds, and integrates the speech recognizer's acoustic and language model probabilities into the understanding framework. By computing the probabilities of the relevant subsets of the sausage, we can also compute the needed priors for the integration of referent probabilities discussed above.

A Synthetic Character that Responds to Speech Commands

We have implemented a machine controlled character for Neverwinter Nights that uses the FUSS presented here to do a player's bidding. During gameplay, we continuously update the physical and intentional situation models as the game progresses, and perform speech recognition, parsing, and binding to the situation models whenever the player speaks. The synthetic character currently only fulfills relatively simple commands such as opening chests and attacking monsters, but it does so robustly and by taking into account the situation. For example, it will interpret the same utterance (e.g. "pull the lever") differently at two different points in time, and understand correctly despite the top speech recognizer hypothesis being the wrong one.

Given the 90 utterances that had noun phrases directly referring to objects in the physical game setting, we asked the question of how often often this character would be able to determine the correct referent using the framework sketched here. Table 1 shows the fraction of correctly determined referents using only the physical bindings, only the plan bindings, and using the integration of both aspects of the situation model into the understanding process. Integrating both aspects of the model clearly goes a long way towards robust disambiguation, even with utterances this noisy. We have already discussed other ways to improve disambiguation performance through more sophisticated physical models and language bindings, and expand on this below.

Physical Only	Plan Recognition Only	Full Model
27/90 (30%)	21/90 (23%)	50/90 (56%)

Table 1: Fraction of referents correctly understood

Figure 5 shows the successful disambiguation of the sausage in Figure 4. The relevant words of the sausage are shown at the top of the figure, followed by a few of the constituents the linguistic parser assigns to them (the full Earley parse contains thousands of constituents). The parser finds the lengthy and highly probable phrase from the sausage shown here, and the physical binding of "gate" produces the highly skewed probability distribution on the left, where the two bars correspond to the two doors in the puzzle. At the bottom of the figure is another partial parse, this time of the event stream. The solidly outlined boxes correspond to the last few events and constituents found, whereas the boxes with dashed outlines are predicted constituents. Thus, the player has just asked for the first chest (chest 4) to be unlocked, and has retrieved the Chest Key from it. It stands to reason that he or she will now attempt to access the second chest to use this key (and acquire the Door Key in the process), and the plan parser properly predicts this. To do so, the player must enter the East room, and the parser thus predicts that he or she will next ask the other player to pull the lever that opens the door. Whether this will be expressed by referring to the lever or the door itself is arbitrary, and thus the probability distribution produced by the plan recognizer at this stage is confused between the two objects as likely

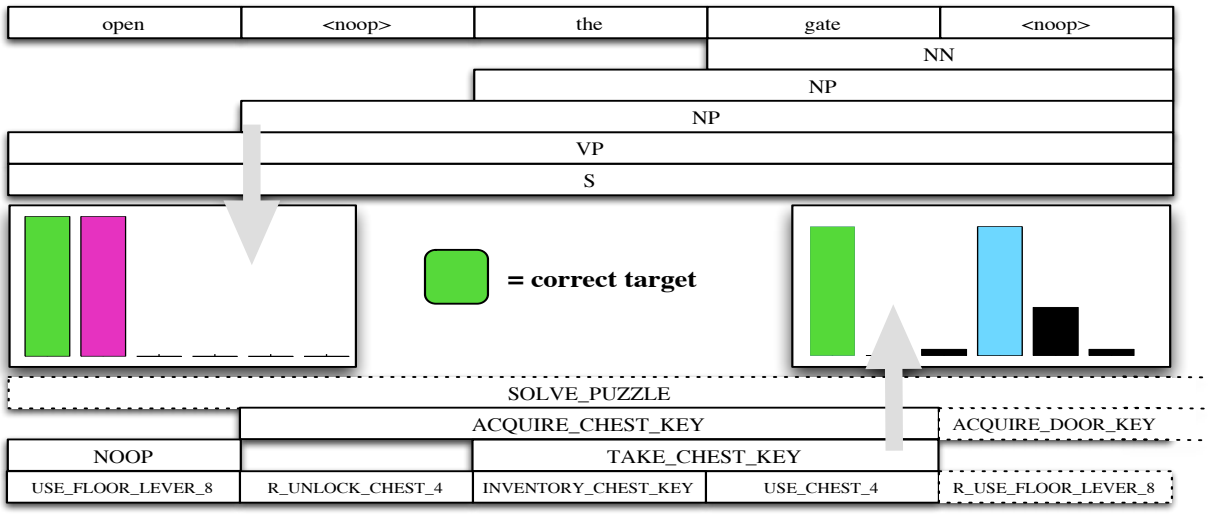

Figure 5: An example linguistic and plan parse fragment showing disambiguation of the sausage from Figure 4: "Can you open the gate again".

referents. Merging the two distributions as described above yields a clear target.

Conclusion

We have argued that natural language and especially speech are the ideal communication channels for interaction with synthetic characters in game environments. We have suggested that games make an ideal platform for investigating frameworks that help such characters understand noisy and spontaneous speech by capturing its ambiguities and resolving them using the current game situation. The presented framework, FUSS, achieves robust referent disambiguation by taking into account both the objects present when an utterance occurs as well as the speaker's current plans. We believe that this approach can be taken much further with more sophisticated situation models and language bindings. For example, players in our study smoothly go from utterances like "pull the lever for me" to "open the door" to "hit me again" to "let me out" (all commanding the other character to perform the same action), a progression touching on the physical and planning realms mentioned here, but also including aspects of spatial confinement and change of language due to shared experience and repetition. We believe that integrating these insights into FUSS will lead to better coverage and more robust performance. Conversely, this research shows that robust situated speech understanding by synthetic characters in games is possible, and will hopefully lead to their deployment in future games.

References

Bioware Inc. 2002. Neverwinter Nights. http://nwn.bioware.com.

Bobick, A. F., and Ivanov, Y. A. 1998. Action recognition using probabilistic parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Earley, J. 1970. An efficient context-free parsing algorithm. *Communications of the ACM* 6(8):451–455.

Gorniak, P. J., and Roy, D. 2004. Grounded compositional semantics for visual scenes. *Journal of Artificial Intelligence Research* 21:429–470.

Hakkani-Tur, D., and Riccardi, G. 2003. A general algorithm for word graph matrix decomposition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of teh 41st Meeting of the Association of Computational Linguistics*.

Mangu, L.; Brill, E.; and Stolcke, A. 1999. Finding consensus among words: Lattice-based word error minimization. In *Proceedings of EUROSPEECH'99*, volume 1, 495–498.

Narayanan, S. 1997. *KARMA: Knowledge-based Action Representations for Metaphor and Aspect*. Ph.D. Dissertation, University of California, Berkeley.

Pynadath, D. V., and Wellman, M. P. 2000. Probabilistic state-dependent grammars for plan recognition. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI2000*. Morgan Kaufmann Publishers.

Schuler, W. 2003. Using model-theoretic semantic interpretation to guide statistical parsing and word recognition in a spoken language interface. In *Proceedings of the Association for Computational Linguistics*.

Stolcke, A. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics* 21(2):165–201.

Yoshida, N. 2002. Utterance segmenation for spontaneous speech recognition. Master's thesis, Massachusetts Institute of Technology.