Automatic Extraction of Spatial Location for
Gesture Generation

by

Yang Gao

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology
February 8, 2002
Copyright 2002 Yang Gao. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
Distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

Author
Department of Electrical Engineering and Computer Science
February 1, 2002

Certified by
Justine Cassell
Thesis Supervisor

Accepted by
Arthur C. Smith
Chairman, Department Committee on Graduate Theses
Automatic Extraction of Spatial Location for Gesture Generation

by

Yang Gao

Submitted to the
Department of Electrical Engineering and Computer Science

February 1, 2002

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

The Behavior Expression Animation Toolkit, BEAT, is a system that adds appropriate and natural-looking nonverbal behaviors for Embodied Conversational Agents, or ECAs. BEAT is designed to plug into any discourse generation module of an ECA and can work with any animation engine to add life to software agents. The BEAT implementation discussed here has the capability of parsing a VRML scene description and generate relative configuration as well as shape conveying gestures automatically. The behavioral output is tightly synchronized with the speech, both contextually and phonologically. Gesture generation takes into account, what the agent is about to say, what the agent has said previously, and the communicative intention of the agent that motivates the current response.

Thesis Supervisor: Justine Cassell
Title: Associate Professor, MIT Media Laboratory
Contents

1. Nonverbal Behaviors .................................................. 4

2. Embodied Conversational Agents .................................. 18

3. The House Description Experiment .............................. 26

4. Beat Version 1.0 .......................................................... 30

5. Improvements upon BEAT v1.0 ................................. 32

6. Integration of MACK and BEAT v2.0 ......................... 54

7. Conclusion and Future Work .................................... 57
1. Nonverbal Behaviors

In today’s fast-paced world, telephony and electronic mail have become pervasive, especially in business and academia. However, although these forms of communication are faster and more convenient, they have not diminished the value of face-to-face conversations. This is because there is a qualitative difference between interacting face-to-face and any other alternate forms of information exchange. Consequently, businessmen still have power-lunches and board-meetings. Academics still fly thousands of miles to attend conferences. Even today, when someone has something of real significance to convey, they still prefer to do it in person.

What makes face-to-face conversation seem more natural and feel more personal are the bodily activities of the interlocutors. Nonverbal behaviors such as facial expressions and hand gestures convey information beyond what is spoken. They reveal the emotional state and thought process of the speaker to the listener, even when the speaker wishes to hide his true feelings. Some psychological studies have concluded that more than 65% of the information exchanged during a face-to-face interaction is expressed through nonverbal channels. [Thalmann, 2000]

Conversations are highly coordinated activities. Nonverbal behavior in conversation is tightly synchronized to the verbal utterance. This is not only true for lip-synch but also for facial expressions, gaze, as well as hand gesticulation. Humans are extremely sensitive to visual cues and any behavior that is out of synchronization with speech is easily detected. The speaker’s conveyance and the listener’s uptake are very
analogous to what Goffman [Donikian, 2000] describes as “traffic codes”. These are collision-avoidance techniques employed by pedestrians. The first of which is that people continuously externalize their intentions by their overall body gestures. The second is a process by which pedestrians selectively gather externalized information from other pedestrians, or what Goffman calls scanning. Externalization and scanning are vital in making walking a relatively low-injury-risk activity and yet pedestrians are rarely aware of doing either one. As with these traffic codes, nonverbal expression and their absorption are usually a subconscious and yet integral part of conversation.

The meaning of what is said is very often colored by the accompanying nonverbal behavior. People are extremely proficient at extracting meaning from subtle variations in the performance of these behaviors. [Cassell et al., 2001] Even the slightest changes in pause length, feedback timing, or gaze direction can significantly alter the message that is conveyed. For example, Bolinger (1989) demonstrated that the exact same string of words could carry very different meanings when spoken with different gestures and facial expressions. [Poggi and Pelachaud, 1999] Everyday utterances like “I don’t know” often take on different interpretations depending on what is conveyed nonverbally.

By the same token, not displaying the correct nonverbal behavior at the right moment can be a source of misunderstanding and confusion. For example, marking an unaccented word with a head nod or an eyebrow raise would create asynchrony between verbal and nonverbal cues. As a result, the listener might not be able to pinpoint the focus
of the utterance. Similarly, having the speaker either always staring directly at the listener or always looking away can inspire a very awkward feeling.

It must be noted however that the overall mixture of behavior and speech is not always monolithic. Sometimes, people do say one thing while expressing another. Numerous researchers have reported that the information conveyed nonverbally can be complementary, redundant, or even contradictory to speech. (Argyle and Cook 1976, Ekman 1979, Kendon 1993, McNeill 1992)

Face-to-face conversation is also more robust and tolerant of noise. The use of verbal and nonverbal channels allows for synergism across multiple modalities. Various studies have demonstrated that redundancy of audio and visual signals can improve speech intelligibility and speech perception. (Bolinger 1989, Harder et al. 1983, Magno Caldognetto and Poggi 1997, Schwippert and Benoit 1997) [Poggi and Pelachaud, 1999] In fact, speech recognition performance can be improved significantly if the audio was accompanied by video input. (Risberg and Lubker 1978) For instance, some phonemes can be difficult to distinguish on the basis of sound alone, such as /m/ and /n/, but easily differentiated when lip-shape is taken into consideration. (Obviously /m/ is denoted by lip closure while /n/ is not.) (Jeffers and Barley 1971). Further more, Rogers found, in his 1978 study, that in noisy situations, humans increase their dependence on access to more than one modality. [Cassell and Stone, 1999]
1.1 Types of Nonverbal Behaviors

There are several different kinds of nonverbal communicative behaviors. These include facial expressions, gazing, and hand gesticulation. (Indeed there are others, such as nodding, eyebrow raises, and posture shifts. However, the discussion of these behaviors will be omitted in this thesis.) In the following section, I shall briefly discuss the corresponding communicative functions of facial expressions and gazing. An extended exploration of gesturing will have a section onto itself since it is the central focus of this thesis.

It is important here to make a note of the distinction between communicative functions and communicative behaviors. Communicative functions, sometimes also known as conversational phenomena, refer to specific internal states of the speaker. These are the intentions of the speaker, which he wishes to impart to his listener, such as yielding the floor. Associated with each communicative function is a set of communicative behaviors that externalize the speaker’s intentions, such as pausing and gazing at the listener to yield the speaking turn.

Facial expressions constitute an important communicative channel and serve several key communicative functions. On the interactional level, they regulate the conversation by facilitating turn exchange. Some facial expressions are synchronized at the verbal level, adding emphasis to certain segments of the utterance. Other facial expressions may even substitute for a word or a string of words. They can also express
attitude toward one’s own speech (irony) or toward the interlocutor (like submission). [Pelachaud, 2000] The face is also the primary channel to express emotion.

Gaze offers another key medium for nonverbal communication. As with facial expressions, gaze is also used to achieve smooth turn-taking and to draw the listener’s attention to a particular word of phrase in the sentence. In addition, gaze can play an important role in deixis. Gaze, sometimes accompanied by a pointing gesture, is frequently used to indicate a given object or a point in space.

1.2 Gestures

Yet another group of frequently occurring nonverbal behaviors during face-to-face conversation is hand gestures. In some discourse contexts it has been found that about 75% of all clauses have gestures of one kind or another associated with them. [Cassell et al., 1994] In fact, gestural accompaniment is so natural to speech that people even gesture while they are speaking on the phone. (Rime 1982) [Cassell, 1998] It is the pervasiveness of hand gestures in conversation and the multitude of key communicative functions which they serve that make gesticulation the focus of this thesis.

One obvious function of gesturing is to disambiguate muddled speech. Interlocutors in a noisy environment have been shown to rely more on gestural cues. (Rogers, 1978) [Cassell, McNeill, and McCullough, 1998] The higher the noise-to-signal ratio, the more dependence there is on gestures. Similar to the other nonverbal behaviors already mentioned, conversational gestures can also serve to regulate the exchange. Such
interactive gestures (as opposed to topical gestures, which are content dependent) have an advantage over verbal means of coordinating and collaborating because the speaker can insert them with minimal interruption to the topical flow. And like facial expressions and gazing, when gestures overlap with speech, emphasis is conveyed. It has been found that almost 50% of utterances for descriptive discourse involve such emphatic gestures. [Cassell et al., 1996]

Our hands are also especially apt for the transmission of certain information. During a conversation, those concepts that are difficult to articulate with spoken text are often conveyed via gestures. (Kendon 1994) [Cassell and Stone, 1999] Hence simultaneity of two distinct events or the relative configurations of two objects in space may be most efficiently expressed by the positioning of the two hands rather than speech. Indeed it is the use of the hands to convey spatial relations that is especially relevant here.

It has also been found that gestures identify the underlying reasoning processes that the speaker did not or could not express verbally. (Church and Goldin-Meadow, 1986 examining Piagetian conversation; Crowder and Newman 1993, examining science explanations) [Cassell, McNeill, and McCullough, 1998] This fact may not be as surprising once we reveal the closeness of gesture generation and the speaker’s mental state. An exploration of this relationship is left to a later part of this section.

So far, we have mostly viewed hand gestures from the speaker’s perspective. Upon first glance, gestures hardly register in the consciousness of the listener. Does
gesturing actually facilitate the uptake of information on the part of the hearer? Experiments by Feyereisen, Van de Wiele, and Dubois (1988) have shown that this is indeed the case. [McNeill, 1992] Gestures can and are tapped by viewers as a source of information.

In a study conducted by Cassell, McNeill, and McCullough, speech and gestures were deliberately mismatched in the narration of a story. [Cassell, McNeill, and McCullough, 1998] That is to say, the information conveyed by speech was different from what is conveyed in gesture. The researchers wanted to see how such conflicts were resolved. The results showed that listeners not only rely on the information conveyed in gesture for comprehension, but sometimes they even ignore what they heard in favor of what they saw. Consequently it is clear that speech and gesture are integral parts of building an unified mental representation in the listener.

The interdependence of speech and gesture is also evidenced by the fact that the listener rarely ever gestures during conversation. Moreover, 90% of all gestures by speakers occur when the speaker is actually uttering something. [McNeill, 1992] Gestures usually end with the speech. Specifically, the hands come to a stop and very often a nonverbal action such as a gaze in the listener's direction yields the floor. [Poggi and Pelachaud, 1999] Therefore it is apparent that the acts of speaking and gesturing are bound to each other in time.
Actually, the tight synchrony necessary between speech and gesture extends beyond just timing of the outputs. Synchronization must also occur in terms of phonology, semantics, and the discourse structure of the conversation. [Cassell, 2000] The phonological synchrony rule states that the stroke of the gesture precedes or ends at, but does not follow, the phonological peak syllable of speech. [McNeill, 1992] The semantic synchrony rule states that if speech and gesture co-occur, they must present the same idea. Finally, the pragmatic synchrony rule states that if speech and gesture co-occur, they perform the same pragmatic functions.

McNeill writes that gestures are a person’s memories and thoughts rendered visible. [McNeill, 1992] As mentioned previously, gesticulation faithfully mirrors the internal state of the speaker. Unconstrained by grammar or convention, gestures are free to display what we imagine in our minds. This must be true when one considers the extreme rarity of “gestural errors” in conversation. [Cassell, 1995] Despite the fact that speech is often full of disfluencies, gestures almost never have false starts or errors. They truly reflect the speaker’s communicative intent. For example, speakers may say “left” and really mean “right”, but they will still point to the right. (McNeill, 1992) Listeners may even correct errors in speech based on what the speaker gestured.

As we have seen, gestures are tightly intertwined with speech not only in time, but also in meaning and function. In fact there is ample evidence to suggest that gesture and speech are generated from the same underlying mental process. For instance we note the temporal synchronization, the parallelism in semantic and pragmatic content, the
simultaneous acquisition in children (Riseborough 1982) [Cassell, McNeill, and McCullough, 1998], and the tendency for both systems to break down in aphasiacs. (Feyereisen 1983, Pedelty 1987) McNeill argues that gestures are an integral part of language, just as much as words and phrases. [McNeill, 1992] He refutes the inclusion of gesture in paralinguistic behaviors because “far from being ‘beside’ language, gesture is actively a part of language.”

1.3 Types of Gestures

Gestures are much more than just random moments of the hands. More often than not, they convey meaning and augment what is expressed in speech. Years of research into human gesticulation patterns have allowed us to group gestures into a few broad categories. Note that these empirical categories are based on generally accepted data and are by no means given as fact.

1.3.1 Emblematics

Emblematic gestures are gestures that can be used in place of speech. Each “emblem” has a specific form that never varies and a specific meaning that is pre-defined. An example of such a gesture is the formation of a “V” with the index and middle fingers. It symbolizes “victory” in America and something entirely different in British culture. Obviously emblems are culture specific. Many more of these emblematic gestures exist in French and Italian cultures than in America. (Kendon, 1993) [Cassell, 2000]
While certainly useful, studies have shown that emblematic gestures rarely constitute more than 10% of the gestures produce during conversation. [Cassell, 1995] This is one of the reasons why emblematiks are left outside the scope of this thesis. Secondarily, emblematiks usually serve to replace speech and therefore they are of little interest since this thesis deals with the generation of gestures that accompany speech.

1.3.2 Propositionals

Like emblematic gestures, propositionals are also conscious displays. They can be useful in certain types of task-oriented talk where the physical world in which the interaction is taking place is also the topic of conversation. [Cassell, 1995] Their interaction with speech is more like the interaction of one grammatical constituent with another than the interaction of one communicative modality with another. While unlike emblematiks in that propositionals are associated with speech, the co-occurring word may be seen as a mere placeholder for the syntactic role of the gesture. Therefore propositionals can not serve many of the communicative functions mentioned in the previous section. Due to their limited functionality propositionals make up a very small percentage of gestures in daily conversation. Consequently, they will also be ignored for the purposes of this thesis.

1.3.3 Spontaneous gestures

Spontaneous gestures are unconscious and unwitting [Cassell, 2000] yet they make up the vast majority of gestures that happen during conversation. This thesis is
especially concerned with the exploration of this type of gestures because they are the key gestural vehicles for expressing communicative intent.

Despite their importance in face-to-face interactions, spontaneous gestures are not consciously accessible, either to the speaker or his interlocutor. Much like the lose of the surface structure of utterances after the extraction of meaning (Johnson, Bransford, and Solomon 1973) [Cassell, 2000], gestures are thrown away immediately after comprehension is achieved. (Krauss, Morrel-Samuels, and Colasante 1991) However, as we have noted previously, the extraction of meaning have been shown to rely heavily upon these gestural cues. This was clearly demonstrated in the gesture-speech mismatch study by Cassell, McNeill, and McCullough. (1999)

Spontaneous gestures are conversational. They are unlike emblems in that they occur in conjunction with speech and that they do not have stereotypic form. [Bavelas et al., 1995] Rather, speakers spontaneously improvise them along with their words and phrases. Yet there is undeniable synchrony between speech and gesture. Also in contrast to emblematic gestures, spontaneous gestures appear to be universal rather than culturally influenced.

What distinguishes a spontaneous gesture from a propositional one is the fact that the former is completely unplanned and unwitting on the part of the speaker while the latter is a conscious motion. As a result, spontaneous gestures occur with much higher frequency and serve many more communicative functions.
1.4 Types of spontaneous gestures

There are four different kinds of spontaneous gestures: iconic gestures, metaphorical gesture, deictics, and beats. They all serve different functions within a conversation. Jointly they help to convey a unified representation of the speaker’s intentions and mental processes. In story telling for instance, underlying narrative structure may be gleaned from the various types of spontaneous gestures employed [Cassell and Stone, 1999]: iconic tend to occur with plot-advancing description of events or objects, deictics often mark the introduction of new characters, and beats indicate the boundaries of episodes. (Cassell and McNeill 1991)

1.4.1 Iconics

Iconic gestures are those spontaneous hand formations that parallel some feature of the event or object being described. [Cassell, 2000] Iconics can specify the manner in which an action is executed, even if this information is not given verbally. Another possible use of an iconic gesture would be to mimic the relative configurations of two objects in question with the positioning of the hands. Iconic gestures are very closely related to the semantic content of speech. They reveal not only the speaker’s internal representation of the event or scene, but also the particular point of view that the speaker has taken towards it. [McNeill, 1992] Therefore, jointly, speech and iconic gestures give a more complete picture of the speaker’s thoughts and intentions.
Further more, McNeill points out that iconics articulate only the relevant features in the context of speaking. [McNeill, 1992] For instance, there cannot be an iconic display of relative configurations if the speaker’s intent is not to provide a description of the scene. In fact, it is argued by McNeill that iconic gestures can not avoid the incorporation of the speaker’s communicative intentions. Iconic gestures are free to show only what is relevant and nothing else. This is exactly the motivation for my interest in iconic gestures.

The unconscious interpretation of iconics is a large and vital part of the uptake of the listener. Again, in the speech-gesture mismatch study (Cassell, McNeill, and McCullough) complementary information that was conveyed only in gesture and not explicitly in speech still found their way into the listener’s retelling of the narrative. [Cassell, McNeill, and McCullough, 1998] Therefore iconic gestures reveal to the listener aspects of the speaker’s inner mental processes which are often omitted in speech.

### 1.4.2 Metaphorics

Another type of spontaneous gesture is metaphoric. Like iconic gestures, these are also pictorial. [Cassell, 2000] However, the concepts they depict are abstract and without physical representation. Therefore the form of a given metaphoric gesture comes from a metaphor of the abstract notion that one wishes to express. This is clearly the reason for its name. Usually it is a visual imagery that in some way represents the concept. For example, metaphoric gestures may signify the start of a new narrative or a new segment or narration. [Cassell, McNeill, and McCullough, 1998]
1.4.3 Deictics

Deictics are spontaneous pointing gestures that spatialize or locate certain aspects of the discourse. [Cassell, 2000] These referent discourse entities could be physically existent or abstract. In fact most pointing gestures in narratives and conversations are of the abstract kind. [McNeill, 1992] However, most human face-to-face interactions involve very few deictic gestures, with the exception of shared-task type discussions. [Cassell et al., 2001]

1.4.4 Beats

Beat gestures are small biphasic movements that, unlike iconics, do not change in form with the content of the accompanying speech. [Cassell, 2000] Their primary function is to mark instances of one’s own linguistic contribution, speech repairs, and reported speech during conversation. Therefore beats index segments of speech that do not advance the plot line of the conversation but rather provides the pragmatic structure. [McNeill, 1992]
2. Embodied conversational agents

Embodiment has been shown to improve the quality of interactions between humans and computer agents. This is hardly surprising since, as has been expounded upon previously in this paper, face-to-face conversation is still the most engaging and natural form of information exchange for humans. Therefore interactive agents that closely mimic this form of communication would be preferable to users.

Koda and Maes [Cassell et al., 2001] and Takeuchi and Naito both found that users rated interfaces with animated faces more engaging and entertaining than faceless functionally equivalents. Similar findings were established by Andre’s animated presentation agent, PPP Persona. Users said that the equivalent interface without the agent was less helpful.

Embodied conversational agents also make better pedagogical tutors. By creating the illusion of life, animated agents may significantly increase the time that children seek to spend with educational software. [Lester et al., 1999] Lester discovered that ECAs in interactive learning environments can produce what he calls the “persona effect”. This is a phenomenon in which the mere presence of a lifelike character can have a measurable positive influence on the student’s perception of their learning experience. (Lester et al. 1997)
2.1 Improving ECAs with Nonverbal Behavior

In the pursuit of natural conversational interfaces, it is not enough to stop at co-presence. Systems with co-presence still suffer from certain problems that lead to the breakdown of the dialog flow. These include user tendencies to needlessly repeat themselves due to the lack of nonverbal feedback by the agent, as well as frequent user interruptions due to confusion in speaking-turn exchange. (Oviatt 1995) [Cassell and Stone, 1999] These problems all stem from the fact that the agent is not utilizing the nonverbal modalities, which, as we have seen, play a key role in conversation.

Embodiment is good but we could do better still. The vital contributions of nonverbal behavior on conversation were carefully documented in the last chapter. It is therefore important to actually integrate all of the modalities available to the agent into a single unified communication scheme. That way, embodiment can serve an even more useful function.

Of course this is easier said than done. Humans are very sensitive to any errors perceived in nonverbal signals. Wrong movements or the lack of synchrony are immediately detected. Therefore, nonverbal behavior generation, when implemented incorrectly, would do more harm than good. Misleading or even conflicting visual signals would be a major source of confusion for users. Pelachaud points out that as synthetic agents are becoming more and more realistic, their human interlocutors are becoming more and more demanding on the quality of animation. [Pelachaud, 2000] Realistic models require equally realistic behaviors.
Studies involving ECAs with some limited communicative behaviors yielded very promising results. Cassell and Thorisson (1999) showed that humans are more likely to consider computer agents more lifelike and to rate their language skills more highly when those agents utilize not only speech but also appropriate nonverbal channels. [Cassell and Stone, 1999] In a similar study involving autonomous conversational avatars, users judged the system as more natural and engaging than a comparable system but without communicative behavior generation. [Cassell and Vilhjalmsson, 1999]

2.2 Gesturing and ECAs

Of especially great interest to me is the generation of spontaneous conversational gestures in interactive agents. One obvious benefit, as mentioned before, is that gestures can convey certain kinds of information better than speech. Our hands are uniquely apt at indicating spatial relationships of objects or the simultaneity of events.

Furthermore we know that, for human interlocutors, spontaneous gestures directly mirror the thought process and communicative intentions of the speaker. Therefore, computer agents that duplicate such spontaneous behaviors are likely to give the impression of cognitive activity and even the existence of a mind [Cassell, 2000], thereby enhancing the believability and realism of the ECA.
2.3 Related Systems

The notion of utilizing multiple modalities for conversational agents is not a new one. Attempts at incorporating speech and gesticulation in ECAs can be traced back to the 1980s. From then to now, measurable progress has been made as system designers strive to make their agents more and more human in behavior. Of course we are still far from achieving one hundred percent human-like conversational behavior in interactive agents, which may very well be an AI-complete problem. However, in the rest of this chapter we will examine a number of domain-specific systems that have made significant strides toward that illusive goal.

2.3.1 Put-that-There

One of the first systems to successfully integrate the verbal and nonverbal output channels is Richard Bolt’s Put-that-There system. (1987) [Cassell, 1995] This system was able to produce both speech and gestures, although integration of these two modalities was quite limited.

As with most of the early attempts at creating gestures to accompany speech, the Put-that-There system focused on using hand movements as replacements for the spoken words. Missing from these systems is the notion of gesture generation based on the underlying discourse structure of the utterance or the communicative intent of the speaker. As we have discussed in previous chapters, most gestures that occur during conversation are not nonverbal replacements for segments of speech. But rather they are spontaneous reflections of thought.
2.3.2 Performative Facial Expressions

Learning from where the “Put-that-There” system fell short, many models have chosen to represent nonverbal behaviors not in terms of the actions involved but rather their communicative function. [Pelachaud, 2000] Indeed, if nonverbal behaviors are to be more than one-to-one substitutes for phrases, then clearly the speaker’s intent and the context of the conversation must both be considered. As we have already noted, different communicative behaviors can, with varying circumstances, express very different meanings and fulfill entirely dissimilar conversational functions.

Poggi and Pelachaud designed a system to generate appropriate facial expressions on the basis of the underlying semantic information. [Poggi and Pelachaud, 1999] Their goal was to construct a natural and expressive human face starting with a meaning-to-face approach, that is to say, a facial simulation directly driven by semantic data.

The input to this system includes propositional content and a general type of goal, such as request, inform, or ask. The system then takes the input and combines it with the known contextual background information. Additional information that is also taken into account for the expression of the performative include the social constraints between the interlocutors and the personality of the speaker.

Even though the primary focus of this thesis is on spontaneous gesture generation rather than the generation of facial expressions, I feel that this system illustrates the
absolute necessity to take both intent and context into consideration when trying to assign appropriate nonverbal behaviors to speech.

2.3.3 Cosmo

An example of a system that focuses on gesture generation is Lester’s Cosmo. The Cosmo system is an animated pedagogical tutor that instructs students on the basics of internet routing. The designers of Cosmo were primarily concerned with achieving deictic believability - to create deictic gestures, motions, and utterances that are both natural and unambiguous. Consequently, Cosmo’s agent behavior planner must be knowledgeable about the physical properties of its world as well as the relative locations of objects in the scene. [Lester et al., 1999] This is what Lester calls the spatial deixis framework.

Along with world knowledge, Cosmo also has a sense of context and communicative intent. In Cosmo, context is represented by a repository of the discourse history. The notion of communicative intent is supplied by the discourse planner to Cosmo’s explanation generator. [Lester et al., 1999] Once the content and structure of the explanation has been determined, the deictic planner is invoked. The three inputs to the deictic planner are a specific communicative act, a topic, and a referent object.

Users of the system unanimously expressed delight in interacting with Cosmo. Most found him fun, engaging, interesting, and helpful. When compared to an “agent-free” version of the learning environment, users unanimously preferred Cosmo.
However, there remains room for improvement. First of all, the behavior planner specifies that the verbal explanation is always initiated when the agent reaches the apex of its pointing gesture. This is deviation from what we know to be the true nature of deictics. Deictics, and spontaneous gestures in general, do not usually precede speech but rather accompany it. Therefore, a more natural, although much more difficult, implementation would be to dynamically bind the occurrence of the deictic to the appropriate word or words in the utterance.

Another criticism of the Cosmo system is its complete separation of speech and gesture generations. The choice of the referring expression and the production of the deictic gesture are implemented as two entirely independent processes. [Cassell and Stone, 1999] This is why the gestures are always redundant to the information already explicit in speech. A purely additive association between the verbal and the nonverbal squarely contradicts McNeill’s notion of a single mental process that is responsible for both speech and gesture. As a result the affordances of the body can not be exploited for the kinds of tasks that it performs better than speech. [Cassell et al., 2001]

Cosmo’s inability to allocate, rather than just duplicate, the communicative load across different modalities is probably not a serious problem in its limited domain. Certainly, for clarification and emphasis, additive redundancy is the desired effect. However, in general, an ECA’s nonverbal modalities can and should be integrated with speech in various other ways as well. Gesturing could provide additional information that
complements speech. Substitution of one signal for another could also be employed. An example of substitution would be using eyebrow raise to mark a nonsyntactically formulated question. [Pelachaud, 2000] In our previous discussion on nonverbal behaviors, we also have noted instances of contradiction from speech. Sometimes people say the opposite of what they mean in order to express irony or sarcasm while allowing their gestures and facial expressions to reveal their true intentions. Therefore, complementation, substitution, contradiction, as well as redundancy should all be in the nonverbal repertoire of a realistic agent.
3. The House Description Experiment

As mentioned in the preceding sections, during a face-to-face conversation, certain modalities are better than others at delivering certain types of informational content. Especially of interest here is the observation that spatial relationships among physical objects are often best expressed with our hands rather than with speech. The focus of my work is to augment the set of appropriate and natural-looking hand gestures available to embodied conversational agents such that gestures depicting spatial configuration could be dynamically generated.

The gesture generation criteria used in my implementation are based entirely on the findings of the “House Description Experiment”. This was a three-month-long study carried out by Hao Yan, a then graduate student in the Gesture and Narrative Language research group of the MIT Media Laboratory [Yan, 2000], and supervised by Professor Justine Cassell. Since the study provides the theoretical rationale behind my work it is worthwhile here to briefly delineate its procedure, summarize its results, and point out its relevance to this thesis.

The House Description Experiment was an exploration into how human interlocutors distribute communicative loads across multiple channels. The primary objective of the study was to categorize the surface-level behaviors and correlate these general semantic features to the underlying intentions of the speaker. If such a mapping were possible then it would justify a rule-based framework for generating appropriate gestures.
The data collection procedure for the study was tedious but straightforward. Human subjects were paired up. One subject watched a 15-minute video that walked through the rooms of a house and then was asked to describe back the salient features of the house to the other subject. It should be noted that house-description was chosen as the experiment domain simply to stimulate more naturally occurring gesticulation on the part of the narrator. The interactions were video taped and later meticulously transcribed, using McNeill’s coding scheme for gestures.

Semantic meanings, such as “left”, “right”, “square”, and “circle”, were then categorized into ten separate semantic features. One is Shape, which is a description of the shape of an object. Another is Size, which related the size of an object. Also, there are Location and Relative Position. Location indicates the location of an object while Relative Position is semantic information regarding the spatial configuration of a group of objects. Existence, i.e. the existence of an entity, is yet another semantic category. To complete the list of features, there are also Action, Path, Direction, Manner, and Impression, although these five are not relevant here.

Since the goal of the study was to examine the distribution of communicative load across multiple modalities, semantic features expressed in both speech and gesture were recorded. If a semantic feature articulated in speech was also acted out in gesture then the gesture was classified as redundant. On the other hand, if the semantic features in gesture and speech did not overlap, then the gesture was classified as complementary. This
working distinction between redundant and complementary gestures agrees with the definitions presented in the second chapter.

Lastly, the speakers’ intentions were inferred from the context of the dialogues. The ontology of the communicative goals mirrored that of the semantic features. Relevant categories are “introduction of a single object”, “introduction of multiple objects”, “describing the configuration of multiple objects”, and “describing the location of an object”. The other categories were “describing an action”, “giving a general impression”, and “enumerating a group of objects”, but, again, these are not important in this context.

For each utterance, we have the associated intention and the semantic feature (or features) expressed, whether in gesture, or in speech, or redundantly in both. Consequently, we can derive reasonably accurate heuristics for assigning features to modalities, as a function of communicative intent. Six such rules were formulated and four of which are implemented in this thesis.

The first rule states that when the speaker wishes to introduce a single object, the existence of the object is conveyed in speech while complementary information regarding its shape and location is usually given in gesture. The second rule states that when introducing multiple objects, Existence and Number are likely features in speech while Relative Position is likely provided in gesture. Since the semantic features in speech and in gesture are different in this case, the gesture would be complementary. The third rule
applies to the description of the configuration of multiple objects, in which case Existence is likely to be found in speech and accompanied by gestures that convey Relative Position and Shape information. The fourth and final heuristic that is relevant here has to do with describing an object’s location. This rule says that the Location feature would be expressed in both speech and gesture. (Note that there are two additional heuristics not mentioned here. These were deemed out of the scope of this thesis. Please refer to the future works section for a brief discussion regarding these two rules.)

These gesture generation heuristics established by the House Description Experiment serve as the empirical foundation for my implementation. In the next section, I will describe the nonverbal behavior generation framework around which my implementation is built. It is called BEAT, which stands for Behavior Expression Animation Toolkit.
4. BEAT Version 1.0

While it would be highly desirable for embodied conversational agents to be able to utilize both verbal and nonverbal modalities, there are two major obstacles that stand in the way. As I have mentioned earlier in the paper, the key issues of concern here are the appropriateness of the gestures generated and their synchronicity with voice output. Humans respond only to nonverbal behavior that is both contextually appropriate and tightly synchronized with the spoken words. The Behavior Expression Animation Toolkit helps ECA’s do just that. [Vilhjalmsson et al., 2001]

From plain text input, BEAT is able to generate synthesized speech as well as the corresponding nonverbal behaviors. These behaviors are the result of real-time linguistic and contextual analysis of the input followed by the application of a set of rule-based generators and filters. BEAT is based on years of extensive research into conversational behaviors. Numerous behavior rules are distilled from hours and hours of video recordings of natural human conversations, in much the same way as the House Description Experiment. In essence, BEAT encapsulates the solutions for both appropriate behavior selection and phonological synchrony in a ready-to-use module.

BEAT version 1.0 was completed in the summer of 2001. This version of BEAT lays out the basic processing pipeline for the system. The design is very modularized to enhance the flexibility of the system. The components of the pipeline are independent and constructed at run-time. This allows the user to customize BEAT with ease. She could mix and match the various generators and filters that come with the system or simply
write her own. BEAT does not need all of its modules to be included in the pipeline in order to run, and indeed, the user is free to leave out various processing stages at her own discretion.

The first stage of the BEAT pipeline is the Language Tagging module and as the name suggests, this is where the utterance is broken up and tagged. A parts-of-speech tagger, an online dictionary (www.wordnet.com) and the contents of the internal Knowledge Base are all utilized to tag the input text, in an XML format. The resultant XML tree is then passed to the Behavior Generation module. This stage consists of two parts, the generators and the filters. The generators add the appropriate behavior tags according to the existing language tags and then the filters delete conflicting gestures in the XML tree. Lastly, the Behavior Scheduling module takes the resultant tree and flattens it, retrieves the timing of the visemes from a text-to-speech engine, and synchronizes the nonverbal behavior with the spoken text.
5. Improvements upon BEAT v1.0

BEAT, version 1.0, takes much of the guesswork and tedium out of animating computational discourse agents. The toolkit plugs right into the output end of most dialogue generation engines and automatically produces the corresponding nonverbal behaviors that are both contextually appropriate and phonologically synchronized. However the true beauty of BEAT is its extensible architecture. While the initial version of this animation toolkit has its limitations, significant amount of improvements could still be made within the design boundaries of its current framework.

5.1 Incorporation of Communicative Intentions

One such improvement is a direct consequence of Hao Yan’s House Description Experiment. Although considerable amount of gesture-generation hints could be gleamed from the output text alone, Yan’s results clearly indicate that nonverbal behavior is often also a function of the speaker’s communicative goals. That is to say, given two identical strings of text, their corresponding nonverbal gestures should vary, depending on the intentions of the speaker. For instance, from the House Description Experiment it was found that gestures indicating relative configuration of multiple objects would often complement their existence and enumeration information conveyed in speech, but only when the intent of the speaker is to introduce those objects. A left hand stroke to the relative position of a table and a right hand stroke to indicate the chair would appropriately accompany the utterance “In the room is a table and a chair.” But only if the speaker is introducing those objects for the first time. Subsequent references to the
The first version of BEAT does not account for communicative intentions and generates nonverbal behavior based only on its textual input. It is necessary then to implement the additional functionality of intent recognition in order to achieve more natural gesture generation.

5.2 Augment the Knowledge Base

Further more, the behavioral repertoire of BEAT is somewhat limited by the content of its Knowledge Base module. This component of BEAT stores the associations between numerous static hand-movements with specific words and thereby extending the set of possible gestures. With version 1.0, the user of BEAT must handcraft much of the expansive-gesture capabilities from within the Knowledge Base. It is a rather time-consuming chore for the user to populate the Knowledge Base, entry by entry. While there certainly should be a repository of special gestures that the BEAT user may wish to include, the number of entries into the Knowledge Base should be kept to a minimum. The more dynamic hand-gestures that could be automatically generated from within BEAT (as opposed to the hard-coded hand-gestures in the Knowledge Base), the better.

5.3 Implementation of BEAT v2.0

The designs and implementation described below address precisely those two flaws of BEAT, version 1.0. Four of the six gesture-generation rules from the House
Description Experiment are added to this new version of BEAT. Obviously, this addition would also necessitate the introduction of communicative intentions into the BEAT processing pipeline. A different kind of knowledge repository, called the Scene Image module, is added to the BEAT architecture. This component will enable BEAT to dynamically generate a huge set of different hand-gestures based on a VRML scene graph. This added feature not just greatly augments the set of gestures available to BEAT, but also perfectly complements the four new gesture-generation rules.

5.3.1 Intention Tags

Due to the highly extensible nature of the BEAT toolkit, adding speaker intentions is very simple. While there might be numerous ways in which speaker intentions could be represented, BEAT's design suggests a very obvious choice. Communicative goals would be translated into intention tags. This ability to define your own tags is the rationale behind choosing eXtensible Markup Language in the first place.

Based on the four rules from Yan’s study, the additional XML tags to BEAT are: `<INTRO_OBJ>`, `<INTRO_OBJS>`, `<DESC_CONFIG>`, and `<DESC_LOCATION>`. These correspond to the four major communicative goals from Yan’s findings, “introduce a single object”, “introduce multiple objects”, “describe the configuration of multiple objects”, and “describe the location of an object”, respectively. With the new intention tags, the input into BEAT is no longer pure text. The intent of the speaker would now also be made available to be processed in conjunction with the text to be spoken.
Typically, an input into BEAT would be the text of an utterance sandwich between a pair of intention tags. For instance, an input could look like “<INTRO_OBJS>There is a huge doorway and a big fireplace.</INTRO_OBJS>”. (Of course, in order to be backward compatible with the earlier version of BEAT, the presence of the intention tags is not required.) It is reasonable to assume that, while previously, the text would be generated by some discourse generator and then passed to BEAT, now, the same discourse generator could provide the intention tags as well. (For further discussion into the generation of intention tags, please refer to the next chapter, on the integration of BEAT and MACK.)

<table>
<thead>
<tr>
<th>Major Communicative Goal</th>
<th>Gesture Type</th>
<th>Semantic Features in Speech</th>
<th>Semantic Features in Gestures</th>
<th>XML Tag Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduce a single object</td>
<td>Complementary</td>
<td>Existence</td>
<td>Shape, Location</td>
<td>&lt;INTRO_OBJ&gt;</td>
</tr>
<tr>
<td>Introduce multiple objects</td>
<td>Complementary</td>
<td>Existence, Number</td>
<td>Relative Position</td>
<td>&lt;INTRO_OBJS&gt;</td>
</tr>
<tr>
<td>Describe the configuration of multiple objects</td>
<td>Complementary</td>
<td>Existence</td>
<td>Relative Position, Shape</td>
<td>&lt;DESC_CONFIG&gt;</td>
</tr>
<tr>
<td>Describe location of an object</td>
<td>Redundant</td>
<td>Location</td>
<td>Location</td>
<td>&lt;DESC_LOCATION&gt;</td>
</tr>
<tr>
<td>Describe a general impression</td>
<td>Redundant (Metaphoric)</td>
<td>Impression</td>
<td>Impression</td>
<td>Not implemented</td>
</tr>
<tr>
<td>Describe the shape of an object</td>
<td>Redundant</td>
<td>Shape</td>
<td>Shape</td>
<td>Not implemented</td>
</tr>
</tbody>
</table>

Table 1. Distribution of semantic features across speech and gesture, as a function of speaker intent.
5.3.2 The Scene Image

Although the addition of intention tags to BEAT is rather straightforward, the generation of the appropriate gestures as called for from the House Description Experiment results is much more complex. The semantic features that must be conveyed in gesture include shape and location of an object or the shapes and/or relative positions of a group of objects. There are no gestures that could be automatically generated by BEAT version 1.0 that could express these features. One possibility is to leave it to the BEAT user to construct a Knowledge Base with all the shape-conveying gestures for all known objects, and all the relative location gestures for all permutations of known objects. Such a task grows exponentially with the number of objects that the user wishes to include. This would not be a pleasant exercise for the user. Unfortunately, with the original BEAT, there is no other alternative.

This in fact is the motivation behind the Scene Image repository. Functionally, it mirrors that of the Knowledge Base, but from the implementation perspective, they are very different. Within the BEAT pipeline, both the Scene Image and the Knowledge Base provide the appropriate gesture tags for Behavior Suggestion, in the Behavior Generation module. However, unlike the static entries of the Knowledge Base, which must be manually configured by the BEAT user, the gestures stored in the Scene Image are dynamically produced from a VRML input file.
5.3.2.1 Building a VRML scene

VRML stands for Virtual Reality Modeling Language. Based on the Open Inventor ASCII format from Silicon Graphics, VRML makes it possible to describe three-dimensional scenes, complete with shapes, lights, textures, and other effects. Although there are other technologies, such as Java3D, for virtual reality rendering, I chose VRML because of three important reasons. First, VRML is a widely accepted web standard and has been around since 1995. Since VRML is intended for the web, it is also attractive because it is very lightweight. VRML files are nothing more than simple text files. For my purposes here, there is no reason for more complexity. Lastly, there are many graphical development tools available for expedient construction of VRML scenes. (SGI’s CosmoWorld is the one I used.)

One could easily build a VRML scene with all the visible objects in their appropriate locations. An example would be a living room, which could contain all the pieces of furniture that the BEAT user can envision, and all laid-out accordingly. Each known object in the living room would be given a name, such as “couch” or “television”. This would be all that the user needs to provide as input to BEAT. The scene graph from the VRML file is then parsed by BEAT and the entries of the Scene Image modules are populated automatically. This is certainly a more appealing alternative to the manual construction of the Knowledge Base.

When a scene graph is parsed, all of the defined nodes from within the VRML file are extracted. Defined nodes are entities in the scene that have non-generic names.
associated with them. Any type of node in VRML can be assigned a name, and therefore defined nodes could be transforms, groups, or simple shape nodes. (For VRML’s programming specifications, please visit the Web3D Consortium, at www.web3d.org.) It is assumed that all objects in the scene of interest to the user would be identifiable with a unique name. In fact, an object’s name is used to retrieve the relevant gestures from the Scene Image repository.

However it is not necessary that the noun phrase used in speech match the exact strings used for the defined-node name. First of all, every word in the utterance is “lemmatized” before attempting a lookup into the Scene Image. This means that words are reduced to their principle forms. For example, “walking”, “walked” both reduce to “walk” in the case of verbs, and “desks”, “desk’s” both reduce to just “desk”. This, in effect, carries out a sub-string match rather than a complete string comparison, and thereby allowing a single entity to be referred to by multiple forms.

Secondly, BEAT attempts to disambiguate similar objects based on their specified features. For instance, in the VRML scene there could be two chairs, and assuming that they were given descriptive names such as “big_soft_chair” and “small_yellow_chair”, BEAT would try to use the accompanying adjectives to distinguish between the two chairs. (Note that the underscore character must be used as separates here because VRML naming convention does not allow white spaces.) So if in speech, one said “the big chair”, or “the soft chair”, or “the big green chair”, they would all map to the VRML object “big_soft_chair”, as opposed to the “small_yellow_chair”. This is because all those
instances would have a match-score of two to "big_soft_chair" while they would only score one for "small_yellow_chair". Of course if there were only one instance of a chair in the VRML scene, then regardless of its descriptive attributes, all "chairs" mentioned in speech would match to that one chair object. Each defined node then corresponds to a unique entry in the Scene Image, and BEAT will attempt to return the best match based on descriptive features of objects, if they are available.

Effectively the Scene Image is a table with three columns and however number of rows necessary to contain each named object. The first column has all the names of the objects in the scene, stored as a lowercased string. This column serves as the primary key during lookup. The second column contains the x-y-z coordinates of the object. The third column contains the most salient shape of the object and its dimensions. These last two pieces of information regarding the object are also parsed from the scene graph.

5.3.2.2 Extracting Location

The actual x-y-z coordinates of any object in the scene are a product of the nested layers of transforms that affect the node. Having the x-y-z coordinates of an object would not only give the absolute location of the object, but also its position relative to all other objects in the scene. Additionally, the x-y-z coordinates of any VRML object are transformed with respect to a given point-of-view before being stored in its Scene Image entry.
This is a convenience feature for BEAT users who would not need to duplicate their VRML scene graphs from multiple perspectives. Obviously, if the BEAT user places the embodied conversational agent to the right of the VRML scene, the relative configuration of the objects would be mirror images of what the agent would see if it had been placed on the left side of it instead. So an object that is to the right of another object from one point of view would be to the left from another perspective. Similar, something within arm’s reach from the front would be all the way across the room when viewed from the back. BEAT takes in the point-of-view parameter and automatically applies the necessary spatial reasoning to a single VRML scene and generates the appropriate locations for the objects. Currently, “front”, “back”, “left” and “right” are the four possible values that the user may specify.

Location and relative position directly correspond to two of the three semantic features necessary for all four of Yan’s gesture-generation rules. The third semantic feature is shape. As with location, shape is also derived from the VRML scene graph. However, determining the shape of an object is not quite as simple as locating the object in space.

5.3.2.3 Extracting Salient Shape

VRML defines three basic shape nodes: box, cylinder, and sphere. If all named objects in any scene were limited to either box, or cylinder, or sphere, then there would be no problem at all. Unfortunately, not only does VRML allow composite shape nodes, which are defined nodes that are made up of several basic shapes, but also complex shape
nodes, which are user-defined complex polygons. What then, should the shape be for a named object that is either composite or complex?

In the complex case, the solution is simple: for all named objects that are user-defined complex polygons, a default shape is stored for that object. This will eventually translate into a default gesture for any and all complex nodes. The question of how to use the hands to depict a complex shape is a very difficult engineering problem and lies outside the scope of this thesis.

In the composite case, we could store the entire arrangement of all the basic shape that make up the object or just pick one. Picking only the one salient shape would be simpler and in fact more human-like. This would be best explained with an example. Image a simple table made up of a flat square as the tabletop, and four long skinny cylinders as legs. The composite object of the flat square and the four cylinders could be called “table”. In most cases, we would describe the shape of the table with a flat open hand, mimicking the flat surface of the tabletop, unless there was something especially unusual about one of the legs. If one of the legs is much longer, or much thinker than the other three, then maybe we would choose to mimic the shape of that leg with our hand instead of the tabletop. Either way, one single salient shape is used to represent the shape of the entire composite entity. It would be hard to image how one could depict the actual shape of the table, which is one flat box with four long cylinders, all with one hand motion.
The heuristic used in this implementation to select the most salient basic shape from a composite object is volume. In other words, for an object made up of several basic shapes, the shape that is the largest in volume would be picked to represent the object as a whole. (The volume comparison is done after all scaling transforms are applied to relevant shape nodes.) The rationale behind this is that the largest shape is usually the most visible part of the object. Since the semantic feature we are concerned with is shape, the most visible shape within an object should be dominant in most cases. Of course there are always exceptions to the rule.

One exceptional case is a composite shape node made up of several basic shapes and one or more complex shape nodes. Since generating gestures that represent complex shapes are not within the scope of this thesis, the volume comparison will only be made among the basic shapes of the composite object. This is reasonable since complex shapes are not handled; yet one could argue that in this scenario, volume might not be a good heuristic for determining saliency. Perhaps it is the complex shape that should be more salient over the basic shapes, regardless of size. After all, the creator of the VRML scene would likely have spent the most time on constructing the complex shape and probably would like to see it emphasized in gesture. Again, since complex shapes only result in a default gesture, there is not much point in giving them priority over the basic shapes in this implementation.

Once the Scene Image is completed, i.e. every named object in the VRML scene has a location and a salient shape, the processing of utterances can begin. Again, to
maintain backward compatibility, any text input without intention tags are process exactly the same way as they would be in BEAT version 1.0. If the presence of either \(<\text{INTRO\_OBJ}>\), or \(<\text{INTRO\_OBS}>\), or \(<\text{DESC\_CONFIG}>\), or \(<\text{DESC\_LOCATION}>\) is detected in the input, a new generator in the Behavior Suggestion module, called the VRMLGestureGenerator, goes to work. (Note that the utterance, with the intention tags, is still processed by the Language Tagging module before being passed to Behavior Suggestion.)

5.3.3 VRMLGestureGenerator

The VRMLGestureGenerator compares all the words in the utterance against the contents of the Scene Image. If a word in the utterance matches some object in the Scene Image, then VRMLGestureGenerator proceeds to retrieve the location and shape information for the object. As mentioned before, the associated information in each object entry of the Scene Image is sufficient to generate all three semantic features: Location, Relative Position, and Shape. All three features are generated in the VRMLGestureGenerator regardless of the actual intent.

This may seem strange and even erroneous. Yan’s study certainly showed that the distribution of features is a function of communicative intent. In the case of “describe location of an object”, only the location feature is expressed in gesture, not the shape of the object. How can it be correct for VRMLGestureGenerator to convey both location and shape without regard to intent?
The functionality of VRMLGestureGenerator is nevertheless correct due to the design rationale behind the two parts of the Behavior Generation module: Behavior Suggestion and Behavior Selection. The Behavior Suggestion is responsible for generating any and all possibly appropriate behaviors. It intentionally over generates. This is the reason for having the second part of the Generation module, where filters in the Selection stage trim down the XML tree that is over laden with excess behavior. Therefore, adhering to this logic, since VRMLGestureGenerator is in the Suggestion stage, it would be correct to over generate.

The x-y-z coordinates of an object is retrieved by VRMLGestureGenerator from the Scene Image and then mapped into the gesture-space of the animated agent. This is necessary since by VRML specifications, all lengths are in meters, while the gesture-space of an agent could be defined by any scale. Therefore, the locations of the objects in a scene must be scaled-down to fit into an imaginary cube in front of the agent, where the agent is able to reach all parts of the cube with either hand. This is the agent’s gesture-space.

In the case of introducing a single object or describing the location of a single object, moving the hand to the location of the object in gesture space constitutes expressing the location feature in gesture. If multiple objects are involved, then the right hand is moved to the location of the first object (the stroke peaking as the name of the object is spoken) and held there until the end of the utterance. Subsequent objects are gestured with the left hand (again, stroke peak coinciding with the name). This way,
relative configuration is expressed, with the right hand serving as a reference point for the
duration of the utterance.

There is, however, one caveat regarding the multiple objects scenario. Sometimes
a direct scaled-down mapping from VRML space to gesture-space would produce points
in the gesture-space that are too close together. This results in awkward-looking gestures,
as the hands merely overlap each other. In order to avoid this, I have introduced a second
step after the direct point-to-point mapping from VRML space to gesture-space. The
additional step “stretches out” the gesture locations of the objects mentioned in the
utterance so as to “fill up” the gesture-space.

Specifically, this fix is implemented as follows. First, the largest difference in
either x, or y, or z coordinates of all objects is found. If that coordinate is the x-axis, then
a scale factor will be applied to all x coordinates such that the leftmost object is pushed to
the leftmost edge of the gesture-space, and the rightmost object is similarly pushed to the
rightmost edge. The same scale factor used for x will also be used for y and z
coordinates, and thereby stretching them out as well. Of course, since x-axis has the
largest difference, and the same scale factor, when applied to y and z coordinates, would
ensure that they remain in the gesture-space. The process is analogous if either y or z-axis
contains the greatest difference.

With the semantic features Location and Relative Position taken care of, the one
remaining feature is Shape. There are a few different ways to correlate hand-gestures to
shape. I chose to implement all three basic shapes, box, cylinder, and sphere with one hand. Obviously shape could also be represented with both hands, but since the movement of both hands should be independent for Relative Position conveyance (recall that the right hand is held in place as reference while the left hand moved to all subsequent object locations), it would be a cleaner implementation to use only one hand for shape depiction.

The shape of a single type of hand shape is used to represent all spherical objects. The palm of the hand faced downward and the fingers are spread out and curved as if grasping a ball. Of course the left and right hand shapes are simply mirror images of each other. Due to the fact that all points on the surface of the sphere are, by definition, equidistant from its center, there can be no variation in the x, y, or z dimensions. Thus all spherical shapes can be adequately expressed with just the hand shape mentioned above. This is not true for boxes or cylinders.

Visibly, boxes can be very different depending on its dimensions. The shape of a box then cannot be generically conveyed by a single hand shape representation. In my implementation, box-shaped objects are divided into three separate categories, each having its own corresponding hand shape. When all three dimensions of the box are roughly the same size, a “cube” hand shape is used. This is the default case. The “cube” hand shape consists of the palm facing outward with the thumb extended straight out and the other fingers closed and bent at the second joints. It is as if the hand was wrapped around a small box, with the fingers conforming to its angular exterior. If one dimension
if much larger than the other two, a "column" hand shape is used. (Note that by "much larger" I mean more than 2.5 times greater in magnitude. 2.5 is a subjective ratio that I felt marked the threshold between "roughly equivalent" and "noticeably elongated".) The "column" hand shape is the same as the "cube", but an up-and-down motion of the hand is used to indicate the elongated dimension. (Of course, had that dimension been in the x direction, the accompanying motion would have been left and right.) Finally, if one dimension is much smaller than the other two (again, the magic ratio of 2.5 is used for this determination), a "plane" gesture is used. This gesture is simply the flat hand with the fingers closed together, and the palm aligned with the smallest dimension of the object. For example, a square tabletop would have the thinnest dimension along the y-axis, and therefore the corresponding gesture would be a flat hand with the palm faced down.

Similarly, cylinders are also divided into three distinct categories. The default case is when the diameter roughly equals the height. A "can" hand shape is used to convey the semantic shape feature. The "can" gesture is very much like the "cube" with the exception that the fingers do not bend at an angle, but instead just curve inward. This is the shape of the hand when it is holding a soda can. A "pole" is when the height is more than 2.5 times greater than the diameter. The "pole" gesture is analogous to the "column" hand shape. The hand forms a "can", and is moved back-and-forth along the axis of the cylindrical object. Finally, when the height is small relative to the diameter, then the shape is a flat plate and the "disk" hand shape would be employed. Currently, the
“disk” is expressed by the circular motion of the index finger, tracing out the circumference of the plate.

This completes the added functionalities to Behavior Suggestion. Now, for every defined object in the VRML scene mentioned in an utterance, all three semantic features, Location, Relative Position, and Shape, can be expressed in gesture. In the Behavior Selection stage of BEAT’s Generation module, filters are applied to trim down some of the behaviors that were over-generated. Specifically when either <INTRO_OBJS> or <DESC_LOCATION> tags are present, the gesture tags that express the Shape features are filtered out of the processing tree. This is done in accordance with Yan’s finding from the House Description Experiment.

The rest of the BEAT pipeline is retained from version 1.0. Because of the extensible design of this toolkit, the Behavior Scheduling module would not need to be altered to accommodate the additions described in this chapter. (However, the compiler, which translates the upper-level behavior tags into animation-specific code, must be updated. As long as the animation engine allows for dynamic arm movements to x-y-z coordinates in the agent’s gesture space, the update should not be too much work. Since the animation engine is not technically part of BEAT, the compiler adjustments will be left to the user.)
5.4 Example Animation

To demonstrate these additional features to the Behavior Animation Toolkit, we will walk through an example utterance in this section. (Note that I will not discuss the processing stages from the original BEAT, such as the Language Tagging, Behavior Scheduling, etc. Please refer to the BEAT v1.0 user manual for details regarding those parts of the toolkit.) In order to invoke the features introduced in the previous chapter, a VRML scene graph must be created. Again, I recommend the use of Silicon Graphics' CosmoWorld application. In Figure 1, we see a very simple room with a table and a fireplace. As is frequently the case, there is not really any need to use overly complex shapes to depict common household objects. Therefore, a quick-and-dirty construction would often suffice.

It is important for the VRML scene to have unique names for object that might be mentioned in speech later on. Descriptive attributes could be given to objects if there are multiple such object types in the scene. In the scene here, there are two walls, and I have chosen to name them “side_wall” and “back_wall” for distinction. Otherwise, all other entities were given single-word names. The construction of this VRML scene took about 5 minutes. Clearly, this is a much more timesaving approach than having to add entries into the BEAT Knowledge Base with location and shape information for each object.
With the VRML scene built and parsed by BEAT, all the necessary information for generating relative-configuration and shape gestures are now stored in BEAT’s Scene Image. In this example we enter the line "<DESC_CONFIG>There is table with a glass of water on it.</DESC_CONFIG>". The presence of the <DESC_CONFIG> tag tells BEAT that the speaker’s intention is to “describe the configuration of multiple objects”. The Language Module output would look something like this.

```xml
<UTTERANCE SPEAKER="AGENT" HEARER="USER">
<DESC_CONFIG>
<CLAUSE>
<THEME>
<ACTION ID="BE">
<W POS="" LEM="there" SYN="RBHD">
    There
```
Recognizing the <OBJECT> entities “table” and “glass” as visible in the Scene Image, the VRMLGestureGenerator generates the appropriate gesture envelopes to enclose them.

From this snippet of BEAT’s Behavior Generation output, we see that the right hand will be used to indicate the table and the left will be used for the glass. The X, Y, and Z attributes of the GESTURE_RIGHT and GESTURE_LEFT tags are the corresponding locations of the table and glass in the agent’s gesture space. Also present in the gesture tags are the HANDSHAPE attributes. Due to their respective dimensions, a flat
horizontal plane has been chosen to represent the table while a “sodaCan” hand shape represents the glass.

The stroke of the gesture for table coincides with the word “table” in the utterance, but, in order to indicate the relative configuration of the table to the glass, the right hand is held in place until the end. The stroke of the glass gesture happens on the word “glass”, and both hands are retracted only at the end of the entire utterance. This can be seen from the Animation Script.

<AnimationScript SPEAKER="AGENT" HEARER="USER">

<START AID="A556" ACTION="GESTURE_RIGHT" X="-0.017372735" Y="0.0" Z="3.5762667E-8" PRIORITY="20" TYPE="VRML" LEFT_HANDSHAPE="y-plane" WI="2" SRT="0.631">
<START AID="A557" ACTION="EYEBROWS" WI="2" SRT="0.631">
<START AID="A559" ACTION="HEADNOD" WI="2" SRT="0.631">
<START AID="A560" ACTION="VISEME" TYPE="E" WI="3" SRT="0.631">
<STOP AID="A557" ACTION="EYEBROWS" WI="4" SRT="1.458">
<START AID="A571" ACTION="GESTURE_LEFT" X="0.052449938" Y="0.04503867" Z="-0.024902333" PRIORITY="20" TYPE="VRML" RIGHT_HANDSHAPE="sodaCan" WI="4" SRT="1.458">
<START AID="A572" ACTION="VISEME" TYPE="A" WI="5" SRT="1.458">
<STOP AID="A556" ACTION="GESTURE_RIGHT" X="-0.017372735" Y="0.0" Z="3.5762667E-8" PRIORITY="20" TYPE="VRML" LEFT_HANDSHAPE="y-plane" WI="4" SRT="1.458">
<STOP AID="A571" ACTION="GESTURE_LEFT" X="0.052449938" Y="0.04503867" Z="-0.024902333" PRIORITY="20" TYPE="VRML" RIGHT_HANDSHAPE="sodaCan" WI="7" SRT="1.999">

The result of the animation is depicted in Figure 2.
Figure 2. "There is a table with a glass of water on it."
6. Integration of MACK and BEAT v2.0

MACK stands for Media-Lab Autonomous Conversational Kiosk. MACK is a software agent embodied as a giant blue cartoon robot. [Huang, 2001] It serves as an interactive guide to the visitors of the MIT Media Laboratory. It has the ability to provide group and project information as well as give directions to interesting places around the lab. MACK is an ideal candidate for a proof-of-concept integration with BEAT for the following reasons.

First of all, MACK is an embodied conversational agent. Therefore it already has the capabilities of discourse generation, which is necessary to provide input into BEAT. In this case, we are not only interested in the generated response text but also the associated intentions. Both of which could be derived from MACK’s discourse generation module. Secondly, MACK attempts to use multiple modalities to communicate with its users but it lacks the framework for automated nonverbal behavior generation. The current version of MACK requires the implementer to manually hard-code all the appropriate behaviors. Clearly MACK could benefit greatly from using a behavior generation toolkit like BEAT. Lastly, Yan’s study that motivated the improvements to BEAT was actually meant for an ECA named REA. (REA stands for Real-Estate Agent, which explains, in part, Yan’s choice of house descriptions as a domain for his experiment.) It was not meant for MACK. However, a successful integration of BEAT and MACK would demonstrate the universal applicability of a behavior generation toolkit to embodied conversational agents in general.
The actual integration of MACK and BEAT was a relatively painless procedure. After all, BEAT was designed to be integrated with ECA systems. The BEAT pipeline essentially replaced MACK’s entire output module. Originally, the Generation Module (GM) produced the appropriate text for MACK to say and passed that to the output stage, where the static gestures were inserted and the utterance was played out. Now, with the addition of BEAT, the GM passes the text directly to BEAT and BEAT takes care of automatically generating the nonverbal behaviors.

The extraction of communicative intentions is carried out in MACK’s Reaction Module (RM), which is the processing stage preceding the GM. The RM is responsible for MACK’s output selection, based on the overall interaction thus far and the most recent user query. This module is implemented as a finite state machine with states such as “Idle”, “Greet”, and “Specific Query Help”. The assignment of abstract response categories in the RM would then translate into concrete text in the GM. Since communicative goals are an abstraction of the actual surface-level output, it would make sense for intention extraction to take place in the RM rather than the GM.

Unfortunately, due to the nature of MACK’s domain, very few states in the RM actually correlated to the four intention tags currently available in BEAT, i.e. <INTRO_OBJ>, <INTRO_OBJS>, <DESC_CONFIG>, and <DESC_LOCATION>. One such match was the RM output of “DIRECTIONS_TO_ROOM”. In fact, any of MACK’s responses that gave directions could correlate to either the <DESC_CONFIG> or the <DESC_LOCATION> tags, depending on how the directions were given.
In the original version of MACK, directions were given mostly in “route-planning” fashion. Route-planning directions usually involve many directional phrase that trace out the actual path. For example, “go straight for thirty yards and then turn left, then go another twenty yards, and turn right …” This would not be very useful here since no named objects were mentioned. Alternatively, directions could be given by referencing landmarks. An example would be, “you will see a big billboard and a house to its left, go down that road until you come up on a bridge …” In this case, relative position of objects becomes vital. Complementary information conveyed in gesture, such as the shape of landmark objects, would also facilitate their recognition.

Therefore, it was necessary to reword much of the direction-giving responses in MACK. I also had to build a VRML scene depicting the interior of the Media Lab, complete with an assortment of salient landmarks. Now with landmark-based directions and an image of the Media Lab, MACK is able to take full advantage of BEAT’s new capabilities for expressing relative configuration and shape information in gesture.

The successful integration of BEAT with MACK demonstrates the usefulness of BEAT to all embodied conversational systems. With BEAT, ECA designers no longer need to worry about producing natural-looking nonverbal behaviors. Even if BEAT integration never becomes as simple as plug-n-play, the amount of tweaking necessary to get BEAT working with a given ECA is orders of magnitude less than independently implementing believable behavior generation.
7. Conclusion and Future Work

Studies on human face-to-face conversations have demonstrated our constant reliance on nonverbal behavior as a communicative channel. In fact, the tight integration of speech and gesture is what sets this method of information exchange apart from email messages or telephone calls, both of which utilize only single modality for expression. Multimodal communication not only provides information in gesture to complement speech but can also allow for synchronous, visual feedback that is non-disruptive to the speaker. Therefore, it is highly desirable for computational dialogue agents to mimic human interlocutors and also adopt multiple channels when conversing with users.

The key to adding natural-looking gestures to speech for any embodied conversational agent is synchrony. Behaviors that are just haphazardly added to the agent’s utterances can distract and even confuse human users. The appropriateness of gestures hinges on both contextual and phonological synchrony. The relevance of a gesture to the current dialogue depends on what the actual utterance is, what was said before, as well as the communicative intention of the speaker. The stroke of the gesture must also coincide, in time, with the correct parts of the text, as the agent utters the speech.

In this thesis, I have introduced the BEAT system that alleviates ECA designers from having to micromanage the agent’s behavioral output. The Behavior Expression Animation Toolkit plugs into the back end of any discourse generation module and produces nonverbal behaviors that are appropriate to the context and integrated with the
speech. Detailed here were the improvements made to the original version of BEAT. Specifically, BEAT's gesture generation is now colored by the speaker's intentions. The incorporation of communicative goals is based on the findings of Hao Yan's House Description Experiment. I implemented four of the six modality distribution rules, enabling BEAT to recognize <INTRO_OBJ>, <INTRO_OBJS>, <DESC_CONFIG>, and <DESCLOCATION>. The semantic features that BEAT is now capable of dynamically generating are location of object, relative configuration of multiple objects, and the salient shape of objects.

All three of these features are parsed from a VRML description of the scene. The combination of gestures that would express relative position as well as shape information for a given utterance is then automatically produced by the new BEAT. This replaces the need to hardcode these behaviors in BEAT's knowledge base and consequently saving much of the configuration overhead that would otherwise be necessary for ECA systems that wish to employ BEAT.

As a proof-of-concept demonstration, this new version of BEAT has been integrated with the MACK system. Using the output from MACK's response generation module, BEAT was able to animate MACK without static gesture tags and thereby entirely replacing MACK's own canned behavior outputs. The ease of the integration process and the success of the result show the usefulness of BEAT for embodied conversational systems in general.
There are several possible additions to BEAT that could still be introduced. One area for improvement would be in how BEAT recognizes and correlates objects mentioned in speech and those described in the VRML scene graph. Currently BEAT does not have the ability to interpret referential expressions. For instance, consecutive utterances like “There is a dresser on the side. On top of it is a small mirror.” It would be desirable for BEAT to understand that the “it” in the second sentence is a pointer for “the dresser”, especially when describing the relative configurations of objects. Clearly, some means of referent resolution would be very beneficial for BEAT. A straightforward approach could involve a history list of noun phrase entities and use a simple recency constraint algorithm to match antecedent. (See “A Simple Model of Anaphora Based on History Lists” by James Allen, 1995.) Of course there are more complex implementations that could also be applicable here.

Along these same lines, BEAT could also utilize a semantic network to correspond objects mentioned in speech with their alternate names. For instance “couch” and “sofa” are synonyms and could both refer to a single entity in the scene. String comparisons alone would not match the occurrence of “couch” in speech to the sofa object in the VRML description. Therefore, some level of semantic insight is necessary. Wordnet is a web-based resource that can provide such relational information for most English words. In fact it is already being used in BEAT for contrast-detection. Similarly, Wordnet could also augment BEAT’s entity matching capabilities.
Another area for future works is in enhancing the gestural repertoire of BEAT. Currently my additions to BEAT allow it to dynamically generate gestures that express relative configuration of objects as well as a limited set of basic shapes. Complex shapes are abstracted into a generic hand gesture. One possible improvement could be to generate hand gestures for complex shapes using their bounding boxes. Although bounding boxes are, by definition, always going to be of the box shape, they can at least tell us the size of the object and its dimensions. The bounding box of an object could be parsed from the VRML scene graph and therefore this addition would fit nicely into the new BEAT structure that I have created. Alternatively, one could also look at the works of Rijpkema and Girard, which deals with generating hand shapes based on the object being gripped.

Finally, more behavior generation rules could be implemented into BEAT, enabling BEAT to produce a wider range of gestures. Studies similar to the House Description Experiment could be carried out, but in various other domains, in order to further deduce patterns in our nonverbal expressions.
References


