Conversation as a System Framework: Designing Embodied Conversational Agents

Justine Cassell, Tim Bickmore, Lee Campbell, Hannes Vilhjálmsson, and Hao Yan

X.1 Introduction

Embodied conversational agents (ECAs) are not just computer interfaces represented by way of human or animal bodies. And they are not just interfaces where those human or animal bodies are *lifelike* or *believable* in their actions and their reactions to human users.

Embodied conversational agents are specifically *conversational* in their behaviors, and specifically humanlike in the way they use their bodies in conversation. That is, embodied conversational agents may be defined as those that have the same properties as humans in face-to-face conversation, including:

- the ability to recognize and respond to verbal and nonverbal input
- the ability to generate verbal and nonverbal output
- the ability to deal with conversational functions such as turn taking, feedback, and repair mechanisms
- the ability to give signals that indicate the state of the conversation, as well as to contribute new propositions to the discourse

The design of embodied conversational agents puts many demands on system architecture. In this chapter, we describe a conversational framework expressed as a list of conversational properties and abilities and then demonstrate how it can lead to a set of *architectural* design constraints. We describe an architecture that meets the constraints, and an implementation of the architecture that therefore exhibits many of the properties and abilities required for real-time natural conversation.

Research in computational linguistics, multimodal interfaces, computer graphics, and autonomous agents has led to the development of increasingly sophisticated autonomous or semi-autonomous virtual humans over the last five years. Autonomous self-animating characters of this sort are important for use in production animation, interfaces, and computer games. Increasingly, their autonomy comes from underlying models of behavior and intelligence rather than simple physical models of human motion. Intelligence also refers increasingly not just to the ability to reason, but also to “social smarts”—the ability to engage a human in an interesting, relevant conversation with appropriate speech and body behaviors. Our own research concentrates on social and linguistic intelligence—“conversational smarts”—and how to implement the type of virtual human that has the social and linguistic abilities to carry on a face-to-face conversation. This is what we call embodied conversational agents.

Our current work grows out of experience developing two prior systems—Animated Conversation (Cassell et al. 1994) and

Ymir (Thórisson 1996). Animated Conversation was the first system to produce automatically context-appropriate gestures, facial movements, and intonational patterns for animated agents based on deep semantic representations of information, but it did not provide for real-time interaction with a user. The Ymir system focused on integrating multimodal input from a human user, including gesture, gaze, speech, and intonation, but was only capable of limited multimodal output in real time.

We are currently developing an embodied conversational agent architecture that integrates the real-time multimodal aspects of Ymir with the deep semantic generation and multimodal synthesis capability of Animated Conversation. We believe the resulting system provides a reactive character with enough of the nuances of human face-to-face conversation to make it both intuitive and robust. We also believe that such a system provides a strong platform on which to continue development of embodied conversational agents. And we believe that the conversational framework that we have developed as the underpinnings of this system is general enough to inform development of many different kinds of embodied conversational agents.

X.2 Motivation

A number of motivations exist for relying on research in human face-to-face conversation in developing embodied conversational agent interfaces. Our most general motivation arises from the fact that conversation is a primary skill for humans, and a very early-learned skill (practiced, in fact, between infants and mothers who take turns cooing and burbling at one another (Trevarthen 1986), and from the fact that the body is so well equipped to support conversation. These facts lead us to believe that embodied conversational agents may turn out to be powerful ways for humans to interact with their computers. However, an essential part of this belief is that in order for embodied conversational agents to live up to their promise, their implementations must be based on actual study of human-human conversation, and their architectures must reflect some of the intrinsic properties found there.

Our second motivation for basing the design of architectures for ECAs on the study of human-human conversation arises from an examination of some of the particular needs that are not met in current interfaces. For example, ways to make dialogue systems robust in the face of imperfect speech recognition, to increase bandwidth at low cost, and to support efficient collaboration between human and machines and between humans mediated by machines. We believe that it is likely that embodied conversational agents will fulfill these needs because these functions are exactly what bodies bring to conversation. But these functions, then, must be carefully modeled in the interface.

Our motivations are expressed in the form of “beliefs” because, to date, no adequate embodied conversational agent platform has existed to test these claims. It is only now that implementations of “conversationally smart” ECAs exist that we can turn to the evaluation of their abilities (see, for example, Nass, Isbister, and Lee, chap. X; Oviatt and Adams, chap. X; Sanders and Scholtz, chap. X).

In the remainder of this chapter, we first present our conversational framework. We then discuss how this framework can drive the design of an architecture to control an animated character who participates effectively in conversational interaction with a human. We present an architecture that we have been developing to meet these design requirements and describe our first conversational character constructed using the architecture—Rea. We end by outlining some of the future challenges that our endeavor faces, including the evaluation of this design claim.

X.3 Human Face-to-Face Conversation

To address the issues and motivations outlined above, we have developed the Functions, Modalities, Timing, Behaviors (FMTB) conversational framework for structuring conversational interaction between an embodied conversational agent and a human user. In general terms, all conversational behaviors in the FMTB conversational framework must support conversational functions, and any conversational action in any modality may convey several communicative goals. In this section, we motivate and describe this framework with a discussion of human face-to-face conversation. Face-to-face conversation is about the exchange of information, but in order for that exchange to proceed in an orderly and efficient fashion, participants engage in an elaborate social act that involves behaviors beyond mere recital of information-bearing words. This spontaneous performance, which so seamlessly integrates a number of modalities, is given unselfconsciously and without much effort. Some of the key features that allow conversation to function so well are

- the distinction between propositional and interactional functions of conversation
- the use of several conversational modalities
- the importance of timing among conversational behaviors (and the increasing co-temporality or synchrony among conversational participants)
- the distinction between conversational behaviors and conversational functions

X.3.1 Interactional and Propositional Functions of Conversation

Although a good portion of what goes on in conversation can be said to represent the actual thought being conveyed, or propositional content, many behaviors serve the sole purpose of regulating the interaction (Goodwin 1981; Kendon 1990). We can refer to these two types of contribution to the conversation as behaviors that have a *propositional* function and behaviors that have an *interactional* function, respectively. Propositional information includes meaningful speech as well as hand gestures and intonation used to complement or elaborate upon the speech content. Interactional information, likewise, can include speech or non-speech behaviors.

Both the production and interpretation of propositional content rely on knowledge about what one wishes to say and on a dynamic model of the discourse context that includes the information previously conveyed and the kinds of reasons one has for conveying new information. Interactional content includes a number of cues that indicate the state of the conversation. They range from nonverbal behaviors such as head nods to regulatory speech such as "huh?" or "do go on."

One primary role of interactional information is to negotiate speaking turns. Listeners can indicate that they would like to receive the turn, for example, by raising their hands into space in front of their bodies or by nodding excessively before a speaker reaches the end of a phrase. Speakers can indicate they want to keep the turn, for example, by keeping their hands raised or by gazing away from the listener. These cues are particularly useful for the speaker when pauses in speech may tempt the listener to jump in.

Turn-taking behavior along with listener feedback, such as signs of agreement or simple "I am following" cues, are good examples of the kind of parallel activity that occurs during face-to-face conversation. Speakers and listeners monitor each other’s behavior continuously throughout the interaction and are simultaneously producing and receiving information (Argyle and Cook 1976) and simultaneously conveying content and regulating the process of conveying content.

X.3.2 Multimodality

We can convey multiple communicative goals via the same communicative behaviors or by different communicative behaviors carried out at the same time. What makes this possible is the fact that we have at our disposal a number of modalities that can overlap without disruption. For example, a speaker can add a certain tone to the voice while raising the eyebrows to elicit feedback in the form of a head nod from the listener, all without interrupting the production of content. The use of several different modalities of communication—such as hand gestures, facial displays, eye gaze, and so forth—is what allows us to pursue multiple goals in parallel, some of a propositional nature and some of an interactional nature. It is important to realize that even though speech is prominent in conveying content in face-to-face conversation, spontaneous gesture is also integral to conveying propositional content. In fact, speech and gesture are produced simultaneously and take on a form that arises from one underlying representation (Cassell, chap. X; McNeill 1992). What gets conveyed through speech and what gets conveyed through gesture are therefore a matter of a particular surface structure taking shape. For interactional communicative goals, the modality chosen may be more a function of what modality is free—for example, is the head currently engaged in looking at the task, or is it free to give a feedback nod?

X.3.3 Timing

The existence of such quick behaviors as head nods, which nonetheless have such an immediate effect on the other conversational participant, emphasizes the range of time scales involved in conversation. While we have to be able to interpret full utterances to produce meaningful responses, we are also sensitive to instantaneous feedback that may modify our production as we go.

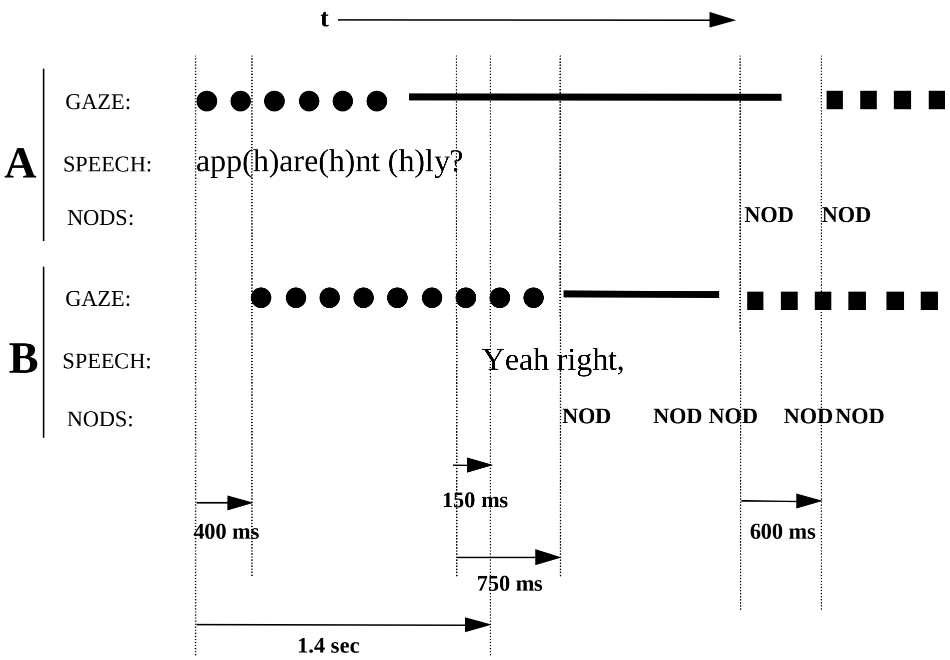

Figure X.1.
A wide variety of time scales in human face-to-face conversation. Circles indicate gaze moving toward other; lines indicate fixation on other; squares are withdrawal of gaze from other; question mark shows rising intonation (from Thorisson 1996, adapted from Goodwin 1981).

In addition, the *synchrony* among events, or lack thereof, is meaningful in conversation. Even the slightest delay in responding to conversational events may be taken to indicate unwillingness to cooperate or a strong disagreement (Rosenfeld, 1987). As demonstrated in figure X.1, speakers and listeners attend to and produce behaviors with a wide variety of time scales. It is remarkable how over the course of a conversation, participants increasingly synchronize their behaviors to one another. This phenomenon, known as entrainment, ensures that conversation will proceed efficiently.

X.3.4 Conversational *Functions* Are Carried Out by Conversational *Behaviors*

Even though conversation is an orderly event, governed by rules, no two conversations look exactly the same and the set of behaviors exhibited differs from person to person and from conversation to conversation. It is the functions referred to above that guide a conversation. Typical discourse functions include conversation invitation, turn taking, providing feedback, contrast and emphasis, and breaking away. Therefore, to successfully build a model of how conversation works, one can not refer to surface features, or conversational behaviors alone. Instead, the emphasis has to be on identifying the fundamental phases and high-level structural elements that make up a conversation. These elements are then described in terms of their role or function in the exchange.

Table X.1 here

This is especially important because particular behaviors, such as the raising of eyebrows, can be employed in a variety of circumstances to produce different communicative effects, and the same communicative function may be realized through different sets of behaviors. The form we give to a particular discourse function depends on, among other things, current availability of modalities such as the face and the hands, type of conversation, cultural patterns, and personal style. For example, feedback can be given by a head nod, but instead of nodding, one could also say "uh huh" or "I see," and in a different context a head nod can indicate emphasis or a salutation rather than feedback. Table X.1 shows some important conversational functions and the behaviors that realize them.

From the discussion above, it should be clear that we make extensive use of the body when engaged in face-to-face conversation. This is natural to us and has evolved along with language use and social competence. Given that this elaborate system of behaviors requires minimal conscious effort, and that no other type of real-time human-to-human interaction, such as phone conversation, can rival face-to-face interaction when it comes to “user satisfaction,” one has to conclude that the affordances of the body in conversation are unique.

The ability to handle natural conversational interaction is particularly critical for real-time embodied conversational agents. Our FMTB conversational framework, then, relies on the interaction among the four properties of conversation described above (co-pursuing of interactional and propositional functions, multimodality, timing, distinction between conversational behaviors and conversational functions). Below, we review some related work before turning to a demonstration of how this model provides a natural design framework for embodied conversational architectures

X.4 Related Work

We have argued that embodied conversational agents must be designed from research on the use and function of the verbal and nonverbal modalities in human-human conversation. Other authors in this volume adhere to this principle to a greater or lesser extent. Other work in interface design has also followed this path in the past, in particular, work in the domain of *multimodal interfaces*. Research on multimodal interfaces has concentrated more on the question of understanding the verbal and nonverbal modalities, whereas embodied conversational agents must both understand and generate behaviors in different conversational modalities. In the sections that follow, we review some previous research in the fields of conversational interfaces and multimodal interfaces before turning to other embodied conversational agent work that resembles our own.

X.4.1 Synthetic Multimodal Conversation

"Animated Conversation" (Cassell et al. 1994) was a system that automatically generated context-appropriate gestures, facial movements, and intonational patterns. In this case, the domain was an interaction between a bank teller and customer. In order to avoid the issues involved with understanding human behavior, the interaction took place between two autonomous graphical agents and the emphasis was on the production of nonverbal behaviors that emphasized and reinforced the content of speech. In “Animated Conversation,” although both turn-taking conversational behaviors and content-conveying conversational behaviors were implemented, no distinction was made between conversational behaviors and the functions they fulfilled. Each function was filled by only one behavior. Because there was no notion of conversational function, the interactional and propositional distinction could not be explicitly made. This was not a problem in for the system, since it did not run in real time, and there was no interaction with a real user, but it made it impossible to extend the work to actual human-computer interaction.

André et al. (chap. X) also implement a system for conversation between synthetic characters for the purpose of presenting information to a human, motivated by the engaging effect of teams of newscasters or sportscasters. Two domains are explored: car sales and "RoboCup Soccer," with an emphasis on conveying character traits as well as domain information. In the car domain, they use goal decomposition to break a presentation into speech acts; and personality and interest profiles in combination with multi-attribute utility theory to organize the presentation of automotive features and values. The result is a sequence of questions, answers, and comments between a seller and one or two buyers. The modalities explored are primarily speech and intonation; although there are some pointing hand gestures. The conversational behaviors generated by this system either fulfill a propositional goal, or convey personality or emotional traits; interactional goals are not considered.

X.4.2 Conversational Interfaces

Nickerson (1976) was one of the pioneers of modeling the computer interface on the basis of human conversation. He provided a list of features of face-to-face conversation that could be fruitfully applied to human-computer interaction, including mixed initiative, nonverbal communication, sense of presence, and rules for transfer of control. His concern was not even necessarily systems that carried on conversations with humans, but rather a model that allowed management and explicit representation of turn taking so the user’s expectations could be harnessed in service of clearer interaction with the computer.

Brennan (1990) argues that human-computer interaction literature promulgates a false dichotomy between direct manipulation and conversation. From observations of human-human conversation, Brennan develops guidelines for designers of both WIMP and conversational interfaces. Key guidelines include modeling shared understandings and provisions for feedback and for repair sequences. The work of both Nickerson and Brennan were essential to our FMTB model.

Badler et al. (chap. X) present a conversational interface to an avatar control task. Avatars interact in the Jack-MOO virtual world, controlled by natural language commands such as “walk to the door and turn the handle slowly.” They developed a Parameterized Action Representation to map high-level action labels into low-level sequences of avatar activity. Humans give orders to their avatars to act and speak, and the avatars may converse with some fully automated characters in the virtual world. Thus, the human interface is effectively command and control, while the multimodal conversation occurs between avatars and automatic characters. No interactional functions such as turn taking are considered in this system. In addition, there is a hard mapping between conversational behaviors and conversational functions, making the use of the different modalities somewhat inflexible.

X.4.3 Multimodal Interfaces

One of the first multimodal systems based on the study of nonverbal modalities in conversation was Put-That-There (1980). Put-That-There used speech recognition and a six-degree-of-freedom space-sensing device to gather user gestural input and allow the user to manipulate a wall-sized information display. Put-That-There used a simple architecture that combined speech and deictic gesture input into a single command that was then resolved by the system. For example, the system could understand the sentence "Move that over there" to mean move the sofa depicted on the wall display to a position near the table by analyzing the position of the pointing gestures of the user. In each case, however, the speech drove the analysis of the user input. Spoken commands were recognized first, and the gesture input only used if the user’s command could not be resolved by speech analysis alone. Certain words in the speech grammar (such as "that") were tagged to indicate that they usually co-occurred with a deictic (pointing) gesture. When these words were encountered, the system analyzed the user’s pointing gestures to resolve deictic references.

Koons extended this work by allowing users to maneuver objects around a two-dimensional map using spoken commands, deictic hand gestures, and eye gaze (Koons, Sparrel, and Thórisson 1993). In his system, nested frames were employed to gather and combine information from the different modalities. As in Put-That-There, speech drove the analysis of gesture: if information was missing from speech, the system would search for the missing information in the gestures and/or gaze. Time stamps united the actions in the different modalities into a coherent picture. Wahlster used a similar method, depending on typed text input to guide the interpretation of pointing gestures (Wahlster 1991).

These examples exhibit several features common to command-and-control-type multimodal interfaces. They are speech-driven, so the other input modalities are only used when the speech recognition produces ambiguous or incomplete results. Input interpretation is not carried out until the user has finished an utterance, meaning that the phrase level is the shortest time scale at which events can occur. The interface only responds to complete, well-formed input, and there is no attempt to use nonverbal behavior as interactional information to control the pace of the user-computer interaction.

These limitations were partially overcome by Johnston (1998), who described an approach to understanding user input based on unification with strongly typed multimodal grammars. In his pen and speech interface, either gesture or voice could be used to produce input and either one could drive the recognition process. Multimodal input was represented in typecast semantic frames with empty slots for missing information. These slots were then filled by considering input events of the correct type that occurred about the same time.

On a different tack, Massaro et al. (chap. X) use nonverbal behavior in Baldi, an embodied character face, to increase the intelligiblity of synthetic speech; they prove efficacy by testing speech readers' recognition rate with Baldi mouthing monosyllables. The output demonstrates improved intelligibility when lip shapes are correct, and the authors have also shown the utility of such a system for teaching spoken conversation to deaf children.

Missing from all these systems, however, is a distinction between conversational behavior and conversational function. This means, in addition, that there can be no notion of why a particular modality might be used rather than another, or what goals are achieved by the congruence of different modalities. The case of multiple communicative goals (propositional and interactional, for example) is not considered. Therefore, the role of gesture and voice input cannot be analyzed at more than a sentence-constituent replacement level.

X.4.4 Embodied Conversational Interfaces

Lester et al. (chap. X) do rely on a notion of semantic function (reference) in order to generate verbal and nonverbal behavior, producing deictic gestures and choosing referring expressions as a function of the potential ambiguity of objects referred to and the proximity of those objects to the animated agent. This system is based on an understanding of how reference is achieved to objects in the physical space around an animated agent and the utility of deictic gestures in reducing potential ambiguity of reference. However, the generation of gestures and the choice of referring expressions (from a library of voice clips) are accomplished in two entirely independent (additive) processes, without a description of the interaction between or function filled by the two modalities.

Rickel and Johnson (1999; chap. X) have designed a pedagogical agent, Steve, that can travel about a virtual ship, guiding a student to equipment, and then using gaze and deictic gesture during a verbal lesson about that equipment. The agent handles verbal interruption and provides verbal and nonverbal feedback (in the form of nods and headshakes) of the student's performance. Although Steve does use both verbal and nonverbal conversational behaviors, there is no way to time those behaviors to one another at the level of the word or syllable. Nonverbal behaviors are hardwired for function: Steve cannot reason about which modalities might be better suited to serve particular functions at particular places in the conversation.

In contrast to these other systems, our current approach handles both multimodal input and output and is based on conversational functions that may be either interactional or propositional in nature. The basic modules of the architecture described in the next section were developed in conjunction with Churchill et al. (chap. X). The architecture grows out of previous work in our research group on the Ymir architecture (Thórisson 1996). In this work, the main emphasis was on the development of a multilayer multimodal architecture that could support fluid face-to-face dialogue between a human and graphical agent. The agent, Gandalf, recognized and displayed interactional information such as gaze and simple gesture and also produced propositional information, in the form of canned speech events. In this way, it was able to perceive and generate turn-taking and back-channel behaviors that lead to a very natural conversational interaction. This work provided a good first example of how verbal and nonverbal function might be paired in a conversational multimodal interface. However, Gandalf had limited ability to recognize and generate propositional information, such as providing correct intonation for speech emphasis on speech output, or a gesture co-occurring with speech. The approach we use with Rea combines lessons learned from both the Gandalf and Animated Conversation projects.

X.5 Embodied Conversational Agent Architecture

The FMTB model described above can be summarized as follows: multiple (interactional and propositional) communicative goals are conveyed by conversational functions that are expressed by conversational behaviors in one or several modalities. This model, which also serves as a strong framework for system design, is lacking in other embodied conversational agents. We have therefore designed a generic architecture for ECAs that derives directly from the FMTB conversational framework described above. We feel that it is crucial that ECAs be capable of employing the same repertoire of conversational skills as their human interactants, both to obviate the need for users to learn how to interact with the agent and to maximize the naturalness and fluidity of the interaction. We believe that in order to enable the use of conversational skills, even the very architecture of the system must be designed according to the affordances and necessities of conversation. Thus, in our design we draw directly from the rich literature in linguistics, sociology, and human ethnography described in the previous section to derive our requirements, based on our FMTB conversational framework.

In general terms, the conversational model that we have described leads to the following set of ECA architectural design requirements:

• Understanding and Synthesis of Propositional and Interactional Information. Dealing with both propositional and interactional functions of conversation requires models of the user's needs and knowledge and the user’s conversational process and states. Producing propositional information requires a planning module to plan how to present multisentence output and manage the order of presentation of interdependent facts. The architecture must include both a static domain knowledge base and a dynamic discourse knowledge base. Understanding interactional information, on the other hand, entails building a model of the current state of the conversation with respect to conversational process (who is the current speaker and who is the listener, has the listener understood the speaker’s contribution, and so on).

• Multimodal Input and Output. Since humans in face-to-face conversation send and receive information through gesture, intonation, and gaze as well as speech, the architecture also should support receiving and transmitting this information and should be modular so that new input and output modalities can easily be added as new technologies are developed.

• Timing. Because of the importance of working with different time scales, and of synchrony among behaviors, the system must allow the embodied conversational agent to watch for feedback and turn requests, while the human can send these at any time through various modalities. The architecture should be flexible enough to track these different threads of communication in ways appropriate to each thread. Different threads have different response-time requirements; some, such as feedback and interruption, occur on a subsecond time scale. The architecture should reflect this fact by allowing different processes to concentrate on activities at different timescales.

• Conversational Function Model. Explicitly representing conversational functions rather than simply a set of conversational behaviors provides both modularity and a principled way to combine different modalities. Functional models influence the architecture because the core modules of the system operate exclusively on functions (rather than sentences, for example), while other modules at the edges of the system infer functions from input and realize functions for outputs. This also produces a symmetric architecture because the same functions and modalities are present in both input and output.

Based on our previous experience with Animated Conversation and Ymir, we have developed an architecture that handles both real-time response to interactional cues and understanding and generation of propositional content. The interactional and propositional functions are capable of being filled by conversational behaviors in several modalities.

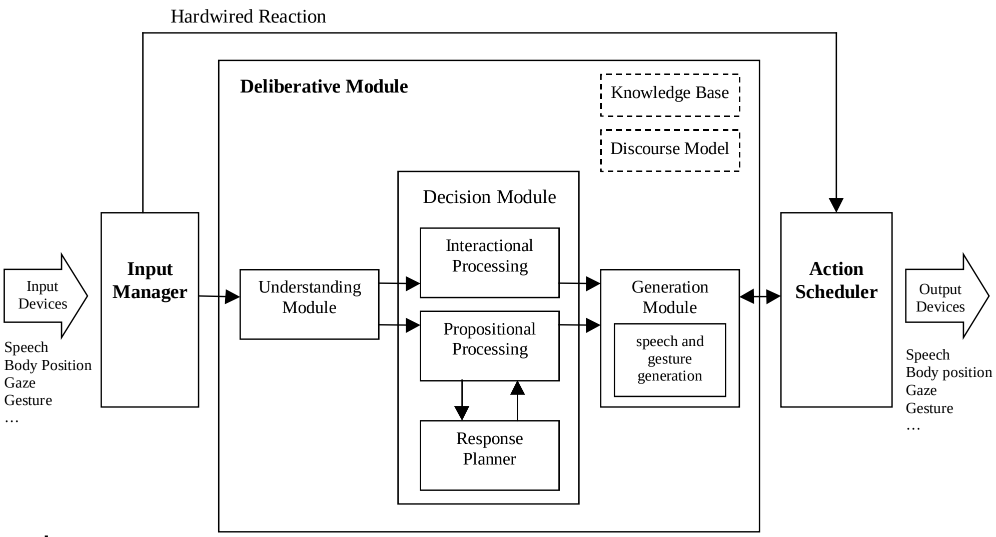

Figure X.2
Overall architecture.

The architecture follows sequential processing of user input (see fig. X.2). First, the Input Manager collects input from all modalities and decides whether the data requires an instant reaction or deliberative discourse processing. Hardwired Reaction handles quick reactions to stimuli such as the appearance or side-to-side movement of the user. These stimuli then provoke a modification of the agent’s behavior without much delay. For example, the agent’s gaze can seamlessly track the user’s movement. The Deliberative Discourse Processing module handles all input that requires a discourse model for proper interpretation. This includes many of the interactional behaviors as well as all propositional behaviors. Last, the Action Scheduler is responsible for scheduling motor events to be sent to the animated figure representing the agent. A crucial function of the scheduler is to prevent collisions between competing motor requests. Each of the modules in the architecture is described next.

X.5.1 Input Manager

In order to support integration of multimodal input from the user, the Input Manager obtains data from the various input devices, converts it into a form usable by other modules in the system, and routes the results to the Deliberative Module. Some interactional information can also be forwarded directly to the Action Scheduler module by way of the Hardwired Reaction module to minimize system response time (e.g., changing the character’s gaze to track a change in the user’s location). The Input Manager will typically receive information from devices that provide speech text, user gesture, location, and gaze information, and other modalities. In all cases, the features sent to the Input

Manager are time-stamped with start and end times in milliseconds.

X.5.2 Hardwired Reactions

Hardwired Reactions enable the character to respond immediately to certain unimodal user inputs that require fast reaction but do not require any inferencing or reference to the discourse model. Examples include tracking the user’s location with the character’s eyes and responding to the user suddenly entering or leaving the interaction space.

X.5.3 Deliberative Module

In order to maintain coherence in the conversation and track the user’s focus, the Deliberative Discourse Processing module maintains a discourse model of the entities introduced in the conversation, the previous statements made by the user and the agent, and other information (e.g., the user’s ultimate and intermediate communicative goals in terms of housing requirements in the real estate domain). The components of this module are grouped together so that they can reference and update these data structures.

The Deliberative Module performs the action selection function of the architecture, which determines what the agent’s contribution to the conversation should be at each moment in time. It receives asynchronous updates from the Input Manager and uses information about the domain (static knowledge base) and current discourse state to determine the conversational action to perform.

The processing is split into three main components: Understanding, Decision, and Generation (see fig. X.2). The Understanding Module is responsible for fusing all input modalities into a coherent understanding of what the user is doing and for translating a set of behaviors into a discourse function, interactional or propositional. It passes these on to the Decision Module in the form of speech acts.

The processing within the Decision Module is split between the processing of interactional communicative acts (those that contribute to the management of the conversational situation) and the processing of propositional communicative acts (those that contribute to the content of the discussion).

The Interactional Processing submodule is responsible for updating the conversational state—namely, whether a conversation with a user has started, who has the turn, and whether the interaction has been put on hold while the user momentarily attends to something else (see fig. X.3). The Propositional Processing submodule is responsible for choosing adequate responses to propositional input (for example, answering questions) and for communicating with the Response Planner if necessary.

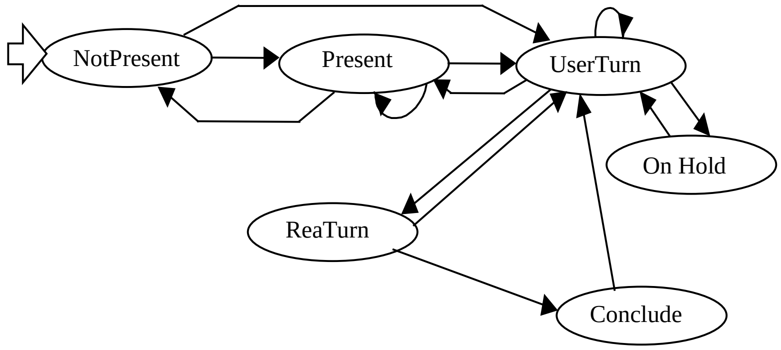

Figure X.3.
Interactional conversational states.

It is important for both the interactional and propositional processes to have access to a common discourse model because interactional information plays a role in validating discourse history updates. For example, the propositional submodule will send off to the Generation Module a speech act to be realized. However, only when the interactional part detects that the agent has successfully concluded an utterance without an interruption from the user does the system consider whether to add the new proposition to the shared knowledge or discourse history.

The Response Planner is responsible for formulating sequences of actions, some or all of which will need to be executed during future execution cycles, to carry out desired communicative or task goals. The Generation Module is responsible for turning discourse functions (such as giving up the turn or conveying a communicative goal) that have been chosen by the Decision Module into actual surface behaviors by producing a set of coordinated primitive actions (such as speech, gesture, facial expression, or a combination of the above) and sending the actions to the Action Scheduler for performance.

X.5.4 Action Scheduling Module

The Action Scheduler is the motor controller for the embodied agent, responsible for coordinating output actions at the lowest level. It takes a set of atomic modality-specific commands and executes them in a synchronized way. This is accomplished through the use of event conditions specified on each output action which define when the action should be executed.

X.5.5 Architecture Summary

In moving from studying conversation between humans to implementing computer systems, we are moving from a rich description of a naturally occurring phenomenon to a parametric implementation. In the process, certain aspects of the phenomenon emerge as feasible to implement, and certain aspects of the phenomenon emerge as key functions without which the implementation would make no sense. The FMTB conversational model is a way of gathering those functions that are essential to the implementation and that can be used as a design framework for the architecture. In the next section, we address the implementation that comes out of an architecture designed in this way.

X.6 Implementation

We are developing an embodied conversational agent within this architecture that is capable of having a real-time face-to-face conversation with a human. The agent, named Rea (for Real Estate Agent), plays the role of a real estate salesperson who interacts with users to determine their needs, shows them around virtual properties, and attempts to sell them a house. We chose the domain of real estate because of the amount of talk with a social function that it provokes (and because we couldn’t pass up the phrase "an experiment in virtual realty"!). That is, in order to put clients enough at ease that they will be able to conceive of parting with the amount of money needed to purchase a house, realtors engage in a lot of talk with a mixed social or interactional function and a propositional function. This domain, then, allowed us to work toward an agent capable of dealing with multiple communicative goals: both interactional and propositional contributions to the discourse, made by way of conversational behaviors carried out in several modalities.

Rea has a fully articulated 3-D graphical body and communicates using both verbal and nonverbal modalities. She is able to describe features of a house using a combination of speech utterances and gestures, and can also respond to users’ verbal and nonverbal input. When the user makes cues typically associated with turn- taking behavior such as gesturing, Rea allows herself to be interrupted, then takes the turn again when she is able. She is able to initiate conversational repair when she misunderstands what the user says. Rea’s speech and gesture output are actually generated in real time from the same underlying knowledge base and description of communicative goals, using the SPUD natural language generation engine (Stone 1998). The following is an excerpt from a sample interaction:

Lee approaches the projection screen. Rea is currently turned side on and is gazing idly about. As Lee moves within range of the two cameras mounted above the screen, Rea turns to face him and says:

REA> Hello. How can I help you?

LEE> I’m looking to buy a place near MIT.

Rea nods, indicating that she is following.

REA> I have a house to show you.

A picture of a house appears on-screen behind Rea who blinks and looks at the house and then at Lee.

REA> It’s in Somerville.

LEE> Tell me about it.

Rea looks up and away while she plans what to say.

REA> It’s big.

Rea makes an expansive gesture with her hands.

Lee brings his hands up as if he is about to speak and so Rea does not continue; instead, she is waiting for him to speak.

LEE> Tell me more about it.

REA> Sure thing. It has a nice garden.

Rea sketches a curved gesture with her hands indicating that the garden extends along two sides of the house.

LEE> How far is it?

REA> It is five minutes to the Porter Square T station.

Rea makes it clear that it is five minutes ***on foot*** *from the T station by making a walking gesture with her fingers.*

LEE> How big is the house?

REA> It has four bedrooms, three bathrooms . . .

Lee interrupts Rea who stops speaking immediately.

LEE> Wait. Tell me, where is the master bedroom?

REA> I'm sorry, I didn't catch that. What did you ask me?

LEE> Where is the master bedroom?

REA> It's upstairs.

Rea points up.

LEE> Where is the master bathroom?

REA> It's next to the bedroom.

Rea brings her hands together to indicate the relationship between the bedroom and the bathroom.

And the house tour continues.

The system currently consists of a large projection screen on which Rea is displayed and in front of which the user stands. Two cameras mounted on top of the projection screen track the user's head and hand positions in space. Users wear a microphone for capturing speech input. A single SGI™ Octane computer runs the graphics and conversation engine of Rea, while several other computers manage the speech recognition and generation and image processing (fig. X.4). The system is implemented in C++ and CLIPS (CLIPS 1994), a rule-based expert system programming language. The modularity of the system design is made possible by using KQML performatives, a common message protocol for interagent communication (Finin and Fritzon 1994), to send and receive messages between different modules.

Figure X.4.
Rea says, "It is next to the bedroom."

In the following sections, we discuss in detail our implementation of the embodied conversational agent architecture in the Rea system. In the discussion of Rea’s implementation, we will follow our discussion of the architecture, moving from the input manager through the discourse processing module to the action scheduler and graphics generation.

X.6.1 Input Sensors

The function of the input manager in the architecture is to handle both verbal and nonverbal inputs from different devices and prepare them for understanding.

In Rea, the input manager currently receives three types of input:

• Gesture Input: STIVE vision software (Azarbayejani, Wren, and Pentland 1996) uses two video cameras to track flesh color and produce 3-D position and orientation of the head and hands at ten to fifteen updates per second.

• Audio Input: A simple audio processing routine detects the onset, pauses, and cessation of speech.

• Grammar-Based Speech Recognition: Speech is also piped to a PC running IBM's ViaVoice98™, which returns text from a set of phrases defined by a grammar.

Data sent to the Input Manager is time-stamped with start and end times in milliseconds. The various computers are synchronized to within a few milliseconds of each other using NTP (Network Time Protocol) clients. This synchronization is key for associating verbal and nonverbal behaviors. Low-level gesture and audio detection events are sent to the Deliberative Module immediately. These events are also stored in a buffer so that when recognized speech arrives, a high-level multimodal KQML frame can be created containing mixed speech, audio, and gesture events. This is sent to the Understanding Module for interpretation.

X.6.2 Discourse Processing

The deliberative processing module is the core part of the architecture. It handles both interactional and propositional facets of the discourse. In Rea, all of the deliberative processing modules are written in CLIPS. Although propositional and interactional elements are considered in an integrated fashion at many points in the system, we will describe them here separately for expository purposes.

X.6.2.1 Interactional Discourse Processing The processing of interactional information in Rea involves some speech but primarily the handling of all non-speech-content inputs and outputs.

The Understanding Module receives a KQML frame from the Input Manager that contains tagged user input, including information from the vision system about the presence or absence of the user and whether he or she is gesturing or not, and information from the audio threshold detector about whether the user has started speaking, has paused, or has finished speaking The Understanding Module looks at the current conversational state (as shown in fig. X.3) and the last known state of all inputs in deciding how to map a particular input into a discourse function. For example, if the user has paused in his or her speaking and the conversational state is UserTurn (user has the floor) and Rea does not take the turn within 0.8 seconds, then a WantingFeedback functional descriptor is created, indicating that the user’s utterance should be acknowledged if possible.

The Decision Module is the center of volition for Rea, since all of its inputs are input discourse functions describing user actions, and its outputs are output discourse functions for Rea to execute. Upon receipt of an interactional message from the Understanding Module, the Decision Module consults the current conversational state and decides on an output action and/or conversational state change. For example, if the conversational state is UserTurn and the Decision Module receives a WantingFeedback message, then a GiveFeedback interactional output message is constructed and sent to the Generation Module for execution, and the state remains UserTurn.

The Generation Module maps requests for output discourse functions into specific output behaviors, based on channel availability, and defines the synchronization requirements for the Action Scheduler to execute. For example, if the interactional output function GiveFeedback is received and Rea's head is not currently being used for a higher-priority behavior, then an Action Scheduler command is generated and sent to cause Rea to nod her head (if her head had been busy, feedback could also have been generated by means of a paraverbal, such as "uh huh").

X.6.2.2 Propositional Discourse Processing The processing of propositional information primarily involves the understanding and processing of speech inputs and the generation of speech and gestural outputs.

As mentioned above, the Understanding Module receives a KQML frame from the Input Manager that contains tagged user input. The Understanding Module's main propositional task is to convert speech input into a valid speech act after resolving referring expressions. The KQML tags from the speech recognizer describe the contents of the utterance and the type of speech act being performed (following Ferguson et al. 1996), in addition to the identification of all discourse entities.

When the Understanding Module has finished binding the discourse entities of the new utterance to existent knowledge base entries, it tries to fill in a speech act template. The template type is chosen according to the incoming speech act tag, but the templates may have preconditions associated with them that have to be fulfilled in order for them to be selected. This way, the choice of template can be sensitive to the discourse model states.

Once the speech act template has been selected and filled in, it is sent to the Decision Module that then needs to evaluate its effect and choose a response. The evaluation may update facts in the dynamic knowledge base and/or create an obligation that the agent needs to attend to. The agent can then perform simple plan reasoning to come up with one or more speech acts to achieve the obligation or communicative goal. The agent commits to the execution of that plan by intending to execute the first speech act of the plan. When it is time to act, the relevant speech act template is filled out and handed to the Generation Module for realization, along with any interactional functions that need to be executed in order to contribute successfully to the conversation.

In Rea, the communicative goal of a speech act can be accomplished by a speech utterance or by the combination of a speech utterance and an appropriate gesture (or gestures). The task of the Speech and Gesture Generation Module is to construct the communicative action that achieves given goals. These propositional goals need to convey domain propositions that encode specified kinds of information about a specified object. The communicative action generated must also fit the context specified by the discourse model, to the best extent possible. We use the SPUD generator ("Sentence Planning UsinDescription") introduced in Stone and Doran (1997) to carry out this generation task.

Figure X.5 shows the structure of the simultaneous speech and gesture generation process in the Generation Module. An utterance generation process starts when the Decision Module sends out a generation speech act. The generation speech act is usually in the "Describe(object, aspect)" form. The request formulator first converts it into a communicative goal that can be understood by the SPUD generator.

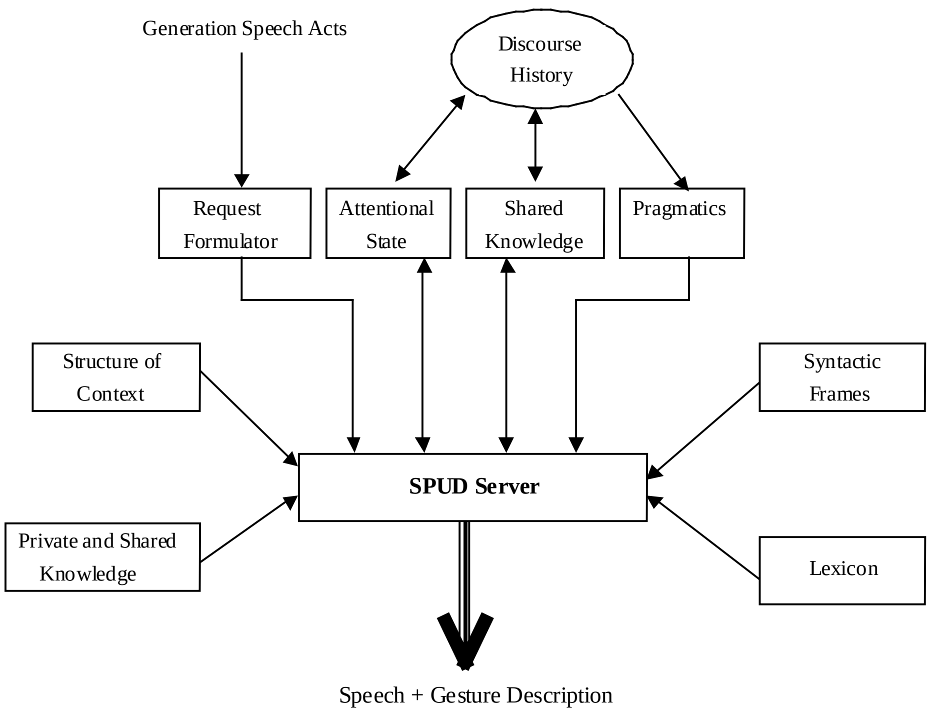

Figure X.5
Speech and gesture generation.

The structure of context, private and shared knowledge, syntactic frames, and the lexicon construct the basic background knowledge base upon which SPUD can draw for its communicative content. The lexical items in speech and constraints on movements in gestures are treated equally as lexicalized descriptors in the knowledge base. The organization of the background knowledge base defines the common ground, in terms of the sources of information that the user and Rea share. It also describes the relationship between Rea’s privately held information and the questions of interest to the user that information can be used to settle. Necessary syntactic and semantic constraints about utterances are also specified in the background knowledge base.

During the conversation, SPUD gets dynamic updates from Rea’s Discourse Model to keep on top of the changing state and context of conversation. These updates include the current attentional state of the discourse (Grosz and Sidner 1986), shared knowledge update to the common ground (Clark and Marshall 1981), and pragmatics by which SPUD looks to prove before an entry can be used.

Based on the communicative goal, background knowledge base, and the updated context of current conversation, SPUD builds the utterance element by element; at each stage of construction, SPUD’s representation of the current incomplete utterance specifies its syntax, semantics, interpretation, and fit to context. If a generation process is successful, a speech utterance along with appropriate gesture descriptions are generated. The gestures generated by the generation process can convey the same piece of meaning that is conveyed by the speech utterances. The use of gestures in this condition will increase the expressiveness and robustness of the communication. The gestures can also complement the speech utterances—namely, they can convey additional information that is not conveyed by the speech utterances. In this case, the communicative load is distributed to both the speech and gestures. The generation process currently uses the combination of the following two kinds of rules to determine whether to generate a complementary or a redundant gesture:

- grouping rules that determine which aspects of an object or an action can be articulated together
- appropriateness rules that determine which aspects/semantics are appropriate or easier to be expressed via the gesture channel, and if appropriate, which gesture can best represent the semantics

Finally, a KQML frame containing the description is sent to the Action Scheduler for execution.

X.6.3 Output System

The multimodal and real-time architectural requirements call for a careful design of the output system. In particular, an embodied conversational agent needs a near-perfect coordination between speech and nonverbal behavior such as gesturing. The slightest mismatch will not only look unnatural, but could in fact convey something different from what was intended. The modularity and extensibility of the architecture require well-defined interfaces between the various components of the output system and have inspired the implementation of a plug-in style motor skill mechanism.

The output system in Rea consists of three main components: a scheduling component, an animation component, and a rendering component. They map into the ECA architecture as Action Scheduler and output devices, respectively. The scheduler receives requests for the activation of various behaviors from the Generation Module. The requests include interdependencies among the behaviors, such as requirements about one behavior finishing before another one starts. The scheduler is therefore responsible for successfully sequencing pending behaviors. The animator assigns a behavior ready to be executed to a motor skill that then becomes responsible for animating the joints of the model by communicating with the renderer (fig. X.6).

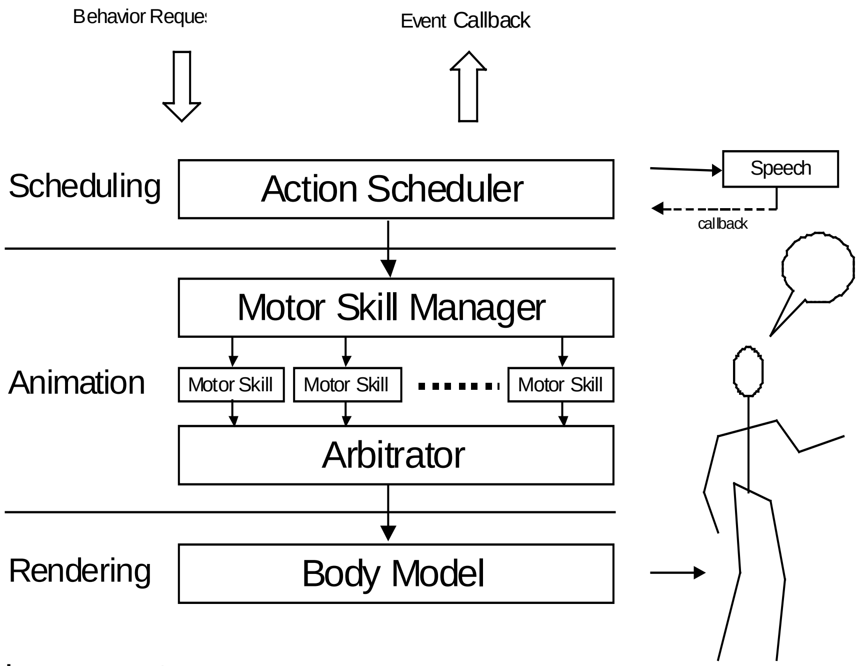

Figure X.6
The three layers of the output system: scheduling, animation, and rendering.

X.6.3.1 Scheduler A behavior description with its preconditions and manner of execution are sent to the Scheduler in a KQML message. The Generation Module typically sends the scheduler a set of behaviors that together, when properly triggered, are meant to carry out a single function, for example an invitation to start a conversation. The scheduler can be instructed to notify the Generation Module through KQML callback messages when certain events occur, such as completion of an output behavior sequence.

Execution of behaviors in the scheduler is event-driven because it is often difficult to accurately predict output behavior execution timings, making it impossible to plan out completely synchronized execution sequences in advance. In addition, some behaviors can produce meaningful events while they are being executed (e.g., the speech synthesis behavior can produce an event after each word is produced) and thus allow other behaviors to be started or stopped when these events occur. Figure X.7 shows an example of an event-driven plan executed by the Action Scheduler with dependencies among the individual behaviors.

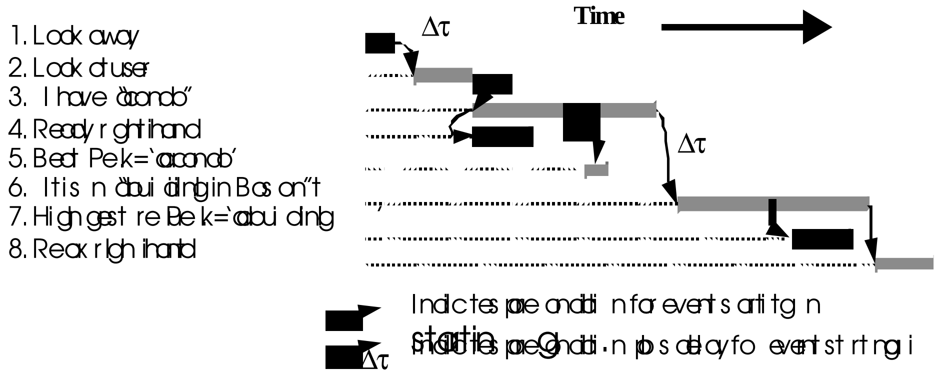

Figure X.7
Example of synchronized speech and gesture output by the Action Scheduler.

The specification sent to the Action Scheduler contains a description of each individual behavior to be executed (a ":content" clause), along with a precondition for the start of the behavior (a ":when" clause) and an optional symbolic label (":id"), which can be used in the preconditions of other behaviors. Figure X.8 shows the KQML input specification for the plan shown in figure X.7.

```
[(action :id H_AWAY :when immediate
         :content (headlook :cmd away :object user))
 (action :id H_AT :when (offset_after :event H_AWAY.END :time 00:01.50)
         :content (headlook :cmd towards :object user))
 (action :id S_CONDO :when (after :event H_AT.END)
         :content (speak :content “I have a condo.”))
 (action :when (after :event S_CONDO.START)
         :content (rightgesture :cmd ready))
 (action  :when (after :event S_CONDO.WORD3)
         :content (rightgesture :cmd beat))
 (action :id S_BLDG :when (offset_after :event S_COND.END :time 00:01.00)
         :content (speak :content “It is in a building in Boston.”))
 (action  :when (after :event S_BLDG.WORD4)
         :content (rightgesture :cmd compose :trajectory vertup :hand bend))
 (action  :when (after :event S_BLDG.END)
         :content (rightgesture :cmd relax))]
```

Figure X.8
Action Scheduler KQML input specification for the plan shown in figure X.7.

The Action Scheduler works by managing a set of primitive behavior objects, each of which represents a set of animations (e.g., "right arm gestures"). When a behavior is commanded to start, it first acquires the body degrees of freedom (DOF) that it requires, such as the set of the right arm and hand joints. It then goes into a starting phase in which it can perform initialization, such as moving the arm into a ready position. Most of the behavior’s actions are carried out in the update phase, which ends when the behavior reaches a natural stopping point, when it is explicitly commanded to stop, or when some other behavior preempts it by grabbing one or more of its DOFs. Before returning to idle, a behavior can go through an ending phase in which it can perform any wrap-up operations needed, such as returning the arm to its rest position.

When the Scheduler has a nonverbal behavior ready for execution, it passes its description over to the animator.

Actions not involving the character's body are executed directly: for example, verbal behavior is sent to the speech synthesizer.

X.6.3.2 Animator The animator checks with the Motor Skill Manager to see if a motor skill capable of handling the request has registered with it. The task of animating joints of the model was broken up into separate motor skills in part because the different skills called for different methods of animation. Motor skills range from straightforward ones, such as those executing a single head nod, to more elaborate ones such as those employing inverse kinematics for pointing at objects or playing key-frame animation. When a motor skill is activated, it asks the Arbitrator for the body DOFs it needs to modify. If two skills ask for the same DOF, the one with the higher priority captures it.

Depending on the implementation of particular skills, the losing skill can keep trying to capture the DOF. This feature is useful for instances where a continuous behavior is momentarily interrupted by an instantaneous one, such as when the character is tracking the user with its gaze and gets asked to glance up and away (higher priority). When the glance is completed, the tracking automatically resumes. The Arbitrator is responsible for keeping track of DOFs in use and allocating them to skills that request them.

All skills can access information about the environment, including virtual objects and the perceived user position through a shared world. Motor skills such as for controlling facing can therefore accept names of objects as parameters.

X.6.3.3 Renderer The rendering engine is abstracted away from the animator by introducing a Body Model layer that essentially maps a DOF name to the corresponding model transformation. We have implemented a Body Model that interfaces with a VRML scene graph rendered using OpenInventor from TGS. The naming of the character's DOFs follows the H-Anim VRML Humanoid Specification for compatibility .

X.7 Evaluation

In this chapter, we have argued that architectures for embodied conversational agents can—indeed must—be built from a model of human-human conversation. And we have provided such a model in the form of a set of properties of human-human conversation that we believe are essential to allowing computers to carry on natural conversations with humans. Note that, following Nickerson (1976), it is important to point out that "an assumption that is not made, however, is that in order to be maximally effective, systems must permit interactions between people and computers that resemble interperson conversations in all respects." Instead, we have argued in this chapter that a successful model of conversation for ECAs picks out those facets of human-human conversation that are feasible to implement, and without which the implementation of an ECA would make no sense.

These claims must be evaluated. To date, empirical evaluations of any kinds of embodied interfaces have been few, and their results have been equivocal. As Shneiderman (1998) points out, ample historical evidence, in the form of a veritable junk pile of abandoned anthropomorphic systems, exists against using anthropomorphized designs in the interface. And Dehn and van Mulken (n.d.), specifically examining evaluations of recent animated interface agents, conclude that the benefits of these systems are arguable in terms of user performance, engagement with the system, or even attributions of intelligence. They point out, however, that virtually none of the systems evaluated exploited the affordances of the human bodies they inhabited: this design paradigm “can only be expected to improve human–computer interaction if it shows some behavior that is functional with regard to the system’s aim.” In other words, embodiment for the sake of the pretty graphics will probably not work.

But note that it is only very recently that embodied conversational agents have been implemented with anywhere near the range of conversational properties outlined above. For this reason, it is only now that we can start to carry out rigorous evaluations of the benefits of conversational embodiment. But evaluation of a system like this takes several forms. We must evaluate the adequacy of the *model* that serves as a design framework; we must evaluate the *implementation* of that design, and we must evaluate the *artifact* that results—that is, we must evaluate the ECA as human-computer interface.

X.7.1 Evaluation of Conversational Model

Our method of evaluating the FMTB conversational model is to look for lacunae in the theory that are pointed out by the implementation. For example, in the earlier system Animated Conversation, interactional and propositional functions were handled entirely separately throughout the system architecture. It was assumed that each utterance had one communicative goal. An unexpected result was that too many head nods and hand gestures were generated, since some performed an interactional and some performed a propositional function. As a result, the current conversational model allows multiple communicative goals for each utterance, of which some may be interactional and some propositional. Our evaluation of the current conversational model, FMTB, has pointed out a weak spot in the understanding of the relationship between conversational behaviors and conversational functions. In particular, it is clear that there is of yet no way of predicting what conversational behaviors will vehicle particular conversational behaviors. That is, we have no theory of the generativity of conversational behaviors from conversational functions.

One particularly difficult arena in which this is true is the generation of hand gestures. We may know that a gesture should convey propositional content, and even that the content should be “a garden that surrounds the house,” and we can autonomously generate these two stages of the production process, but we have no way of predicting what shape of the hands or movement of the hands will best represent this content. For the moment, we resolve this lacuna by providing a list of conversational behaviors. We hope in the future to have a more principled method of solving the problem. We might look at this issue as being one of the *morphology* of conversational behaviors, and we see it as a topic of future research for our group.

X.7.2 Evaluation of Implementation

Our method of evaluating the implementation is simply to see what aspects of the architecture, and of the model before it, are not translated into system behaviors. And, what aspects are badly or imperfectly translated. In this evaluation, one aspect of the FMTB conversational model is strikingly difficult to implement, and that is the feature of *timing*. In fact, our evaluation of our own current implementation points out several weaknesses with respect to timing and to synchrony. First of all, with respect to speed, the natural language generation engine is not currently fast enough to provide any sense of entrainment to human users. That is, users get a sense that Rea is thinking too long before she speaks. Because we have implemented a deliberative discourse processing module and a hardwired reaction module to handle different time scales, this slowness is all the more noticeable. Sometimes Rea reacts instantly, and sometimes she takes too long. Next, with respect to synchrony, we have not yet resolved the issue of how to time gestures perfectly with respect to the speech that they accompany. Thus, for example, hand gestures may occur somewhat after the speech with which they are generated. This simply gives the impression that the system is not working correctly, or that Rea is a bit dim. The problem is due primarily to the difficulty of synchronizing events across output devices, and of predicting in advance how long it will take to execute particular behaviors. That is, it is difficult to predict—and synchronize—the timing of speech synthesis produced by a text-to-speech engine and graphical representations of hand movements produced by a rendering engine.

In order to address this problem, we are currently looking at other text-to-speech engines that may give us phoneme timings in advance, which might facilitate predicting how long it will take to utter a particular phrase. However, a more profound solution, and one that is more in line with the conversational FMTB model presented here, is to endow the Action Scheduler with more intelligence about issues of timing and synchrony. That is, we might conceive of an Action Scheduler that doesn't allow missychronized behaviors to be generated, or that works with other kinds of timing and sychronization constraints. This is a topic for future research.

X.7.3 Evaluation of Interaction

We evaluate the quality of Rea as interface by having her interact with untrained users. Of course, an entirely free interaction with a user would allow us to know whether Rea is ready for prime time (the real estate market) but not allow us to pinpoint the source of any difficulties users might have in the interaction. Therefore, as Nass, Isbister, and Lee (chap. X) describe, we evaluate the performance of our embodied conversational agent through a series of Wizard of Oz experiments where we manipulate one or two variables at a time. Comparing one of Rea’s ancestors (see Cassell and Thórisson 1999 for further details and citations) to an identical body uttering identical words, but without nonverbal interactional behaviors, we found that users judged the version with interactional behaviors to be more collaborative and more cooperative and to exhibit better natural language (even though both versions had identical natural language abilities). On the other hand, performance on the task was not significantly different between the groups. An evaluation of one of Rea’s cousins—a 3-D graphical world where anthropomorphic avatars autonomously generate the conversational behaviors described here—did show positive benefits on task performance. And users in this study preferred the autonomous version to a menu-driven version with all of the same behaviors (Cassell and Vilhjálmsson 1999).

Currently, we are conducting an evaluation that compares (a) face-to-face conversation with Rea to conversation over the telephone with a dialogue system, and (b) whether the user believes that the system (either Rea or the dialogue system) is autonomous to whether it is being manipulated by a human in real time. We will look at the effect of these conditions on users' perception of the system but also on their efficiency in carrying out a task and their performance on that task. In this way, we hope to begin to evaluate the particular conversational properties that make up our FMTB conversational model.

X.8 Conclusions

One of the motivations for embodied conversational agents—as for dialogue systems before them—comes from increasing computational capacity in many objects and environments outside of the desktop computer—smart rooms and intelligent toys, in environments as diverse as a military battlefield or a children's museum—and for users as different from one another as we can imagine. It is in part for this reason that we continue to pursue the dream of computers without keyboards that can accept natural untrained input. In situations such as these, we will need robustness in the face of noise, universality and intuitiveness, and a higher bandwidth than speech alone. We will need computers that untrained users can interact with naturally. And we believe that this naturalness of interaction can come from systems built on the basis of a strong model of human conversation.

In this chapter, we have argued that architectures for embodied conversational agents need to be based on a conversational model that describes the functionality, properties and affordances of human face-to-face conversation. The qualitative difference in architectures designed in this way is that the human body enables the use of certain communication protocols in face-to-face conversation. The use of gaze, gesture, intonation, and body posture play an essential role in the proper execution of many conversational behaviors—such as conversation initiation and termination, turn taking and interruption handling, and feedback and error correction—and these kinds of behaviors enable the exchange of multiple levels of information in real time. People are extremely adept at extracting meaning from subtle variations in the performance of these behaviors; for example, slight variations in pause length, feedback nod timing, or gaze behavior can significantly alter the message a speaker sends.

Of particular interest to interface designers is that these communication protocols come for "free" in that users do not need to be trained in their use; all native speakers of a given language have these skills and use them daily. Thus, an embodied interface agent that exploits them has the potential to provide a higher bandwidth of communication than would otherwise be possible. However, the flip side is that these communications protocols must be executed correctly for the embodiment to bring benefit to the interface.

We believe that Rea begins to demonstrate those correct communications protocols that will make embodied conversational agents successful as human-computer interface.

Notes

Research leading to the preparation of this article was supported by the National Science Foundation (award IIS-9618939), AT&T, Deutsche Telekom, and the other generous sponsors of the MIT Media Lab. Sincere thanks to Kenny Chang, Joey Chang, Sola Grantham, Erin Panttaja, Jennifer Smith, Scott Prevost, Kris Thórisson, Obed Torres, and all of the other talented and dedicated students and former students who have worked on the Embodied Conversational Agents project. Many thanks also to colleague Matthew Stone for his continued invaluable contribution to this work. Finally, thanks to Jeff Rickel and Elisabeth André for helpful comments on an earlier draft.

1. This architecture has been developed in conjunction with the Conversational Characters project at FX Palo Alto Laboratory Inc.

References

Argyle, M., and M. Cook. 1976. *Gaze and mutual gaze*. Cambridge: Cambridge University Press

Azarbayejani, A., C. Wren, and A. Pentland. 1996. Real time 3-D tracking of the human body. In *Proceedings of IMAGE'COM 96* (Bordeaux, France), May.

Bolt, R. A. 1980. Put-That-There: Voice and gesture at the graphics interface. *Computer Graphics* 14(3):262–270.

Brennan, S. 1990. Conversation as direct manipulation: An iconoclastic view. In B. Laurel, ed., *The art of human-computer interface design*, 393–404. Reading, Mass.: Addison-Wesley.

Cassell, J., and K. Thórisson. 1999. The power of a nod and glance: Envelope vs. emotional feedback in animated conversational agents. *Journal of Applied Artificial Intelligence* 13(3):519–538.

Cassell, J., and H. Vilhjálmsson, H. 1999. Fully embodied conversational avatars: Making communicative behaviors autonomous. *Autonomous Agents and Multi-Agent Systems* 2:45–64.

Cassell, J., C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone. 1994. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Computer Graphics*, 413–420. New York: ACM SIGGRAPH

Clark, H. H., and C. R. Marshall. 1981. Definite reference and mutual knowledge. In A. K. Joshi, B. L. Webber, and I. Sag, eds., *Elements of discourse understanding*, 10–63. Cambridge: Cambridge University Press.

CLIPS. 1994. *Reference Manual 6.0* (Technical Report Number JSC-25012). Houston, Tex.: Software Technology Branch, Lyndon B. Johnson Space Center.

Dehn, D., and S. v. Mulken. N.d. The impact of animated interface research: A review of empirical research. *Human-Computer Studies*. Forthcoming.

Ferguson, G., J. Allen, B. Miller, and E. Ringger. 1996. The design and implementation of the TRAINS-96 System: A prototype mixed-initiative planning assistant (Technical Note 96-5). University of Rochester, Department of Computer Science.

Finin, T., and R. Fritzon. 1994. KQML as an agent communication language. Paper presented at the Third International Conference on Information and Knowledge Management (CIKM ’94), Gaithersburg, Maryland, November.

Goodwin, C. 1981. *Conversational organization: interaction between speakers and hearers*. New York: Academic Press.

Grosz, B., and C. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3):175–204.

Johnston, M. 1998. Unification-based multimodal parsing. *Proceedings of COLING-ACL 98*, 624–630. Montreal: Morgan Kaufman Publishers.

Kendon, A. 1990. The negotiation of context in face-to-face interaction. In A. Duranti and C. Goodwin, eds., *Rethinking context: Language as interactive phenomenon*, 323–334. New York: Cambridge University Press.

Koons, D. B., C. J. Sparrel, and K. R. Thórisson. 1993. Integrating simultaneous input from speech, gaze, and hand gesture. In M. T. Maybury, ed., *Intelligent multimedia interfaces*. Cambridge, Mass.: AAAI Press/MIT Press.

McNeill, D. 1992. *Hand and mind: What gestures reveal about thought*. Chicago: The University of Chicago Press.

Nickerson, R. S. 1976. On conversational interaction with Computers. In R. M. Baecker and W. A. S. Buxton, eds., *Readings in human computer interaction*, 681–693. Los Altos, Calif.: Morgan Kaufman.

Rickel, J., and W. L. Johnson. 1999. Animated agents for procedural training in virtual reality: Perception, cognition and motor control. *Applied Artificial Intelligence* 13:343–382.

Rosenfeld, H. M. 1987. Conversational control functions of nonverbal behavior. In A. W. Siegman and S. Feldstein, eds., *Nonverbal behavior and communication*,2d ed., 563–601. Hillsdale, N.Y.: Lawrence Erlbaum Associates.

Shneiderman, B. 1998. *Designing the user interface: strategies for effective human-computer interaction*, 3d ed. Reading, Mass.: Addison-Wesley.

Stone, M. 1998. Modality in dialogue: Planning, pragmatics, and computation. Unpublished doctoral dissertation, Department of Computer and Information Sciences, University of Pennsylvania, Philadelphia.

Stone, M., and C. Doran. 1997. Sentence planning as description using tree adjoining grammar. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, 198–205, Madrid, Spain.

Thórisson, K. R. 1996. *Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills.* Unpublished doctoral dissertation, Department of Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Trevarthen, C. 1986. Sharing makes sense: Intersubjectivity and the making of an infant's meaning. In R. Steele and T. Threadgold, eds., *Language topics: Essays in honour of M. Halliday*, vol. 1, 177–200. Amsterdam: J. Benjamins.

Wahlster, W. 1991. User and discourse models for multimodal conversation. In J. W. Sullivan and S. W. Tyler, eds., *Intelligent user interfaces*, 45–67. Reading, Mass.: Addison-Wesley.