An Architecture for Embodied Conversational Characters

J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjálmsson, H. Yan

Gesture and Narrative Language Group
MIT Media Laboratory
E15-315
20 Ames St, Cambridge, Massachusetts
+1 617 253 4899
{justine, bickmore, markb, elwin, tetrion, hannes, yanhao}@media.mit.edu

Abstract

In this paper we describe the computational and architectural requirements for systems which support real-time multimodal interaction with an embodied conversational character. We argue that the three primary design drivers are real-time multithreaded entrainment, processing of both interactional and propositional information, and an approach based on a functional understanding of human face-to-face conversation. We then present an architecture which meets these requirements and an initial conversational character that we have developed who is capable of increasingly sophisticated multimodal input and output in a limited application domain.

Introduction

Research in computational linguistics, multimodal interfaces, computer graphics, and autonomous agents has led to the development of increasingly sophisticated embodied conversational characters over the last five years.

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 non-verbal input
- The ability to generate verbal and non-verbal output.
- The use of conversational functions such as turn taking, feedback, and repair mechanisms.
- A performance model that allows negotiation of the conversational process, and contributions of new propositions to the discourse.

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 automatically produce context-appropriate gestures, facial movements and intonational patterns for animated agents based on deep semantic representations of information, but did not provide for real-time multimodal interaction with a user. The "Ymir" system focused on integrating multimodal input from a human user, including gesture, gaze, speech, and intonation, and producing multimodal output in real time in an animated character called "Gandalf".

We are currently developing a conversational character 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 will provide 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.

Motivation

As shown in Figure 1, human face-to-face conversation is a complex phenomenon involving understanding and synthesis across multiple modalities and time scales. Speech, intonation, gaze, and head movements function not just in parallel, but interdependently.

Conversational functions, such as turn-taking and feedback, rely on integration of information from all of these channels together. When we attempt to construct

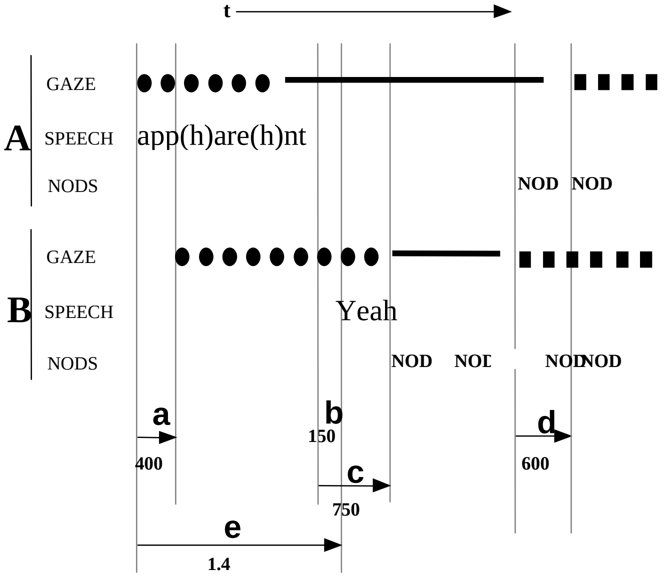

Figure 1: Multi-threaded multimodal behavior in human conversation (Circles indicate gaze moving towards other, lines indicate fixation on other, squares are withdrawal of gaze from other, question mark shows rising intonation) (from (Goodwin 1981), adapted from (Thórisson 1996))

embodied conversational characters which can participate in this kind of interaction, we find that these features have significant ramifications on the design of the characters' control architecture.

There are a number of motivations for developing conversational character interfaces, including:

Intuitiveness. Conversation is an intrinsically human skill that is learned over years of development and is practiced daily. Conversational interfaces provide an intuitive paradigm for interaction, since the user is not required to learn new skills.

Redundancy and Modality Switching: Embodied conversational interfaces support redundancy and complementarity between input modes. This allows the user and system to increase reliability by conveying information in more than one modality, and to increase expressiveness by using each modality for the type of expression it is most suited to.

The Social Nature of the Interaction. Whether or not computers look human, people attribute to them human-like properties such as friendliness, or cooperativeness (Reeves and Nass 1996). An embodied conversational interface can take advantage of this and prompt the user to naturally engage the computer in human-like conversation. If the interface is well-designed to reply to such conversation, the interaction may be improved.

In this paper we will first present a summary of the salient features of human face-to-face conversation, and how these drive the design of an architecture which is able to control an animated character who participates effectively in this kind of interaction. We then present an architecture that we have been developing to meet these requirements and describe our first conversational character constructed using the architecture – Rea.

Human Face-to-Face Conversation

Embodied conversation relies on a number of different *modalities* such as speech, prosody, hand gestures, facial expression and head movements. The speaker employs these channels in parallel, combining modalities as needed for appropriate elaboration, while the listener simultaneously produces multi-modal feedback and contentful responses in a similar way. The speaker and listener accomplish the switching of roles through a sequence of overlapping turn-taking behaviors where the parallel nature of the communication channels, and short timescales of the relevant behaviors provide a seamless transition.

The behaviors that directly contribute to the content delivery or the organization of the conversation are termed *conversational behaviors* and are the surface form of the exchange. Typical conversational behaviors include head nods, glances to the side, raising eyebrows, and speaking. To better understand and explain the relatively complex patterns of conversational behavior, it is important to identify the *functions* that these conversational behaviors serve. Typical discourse functions include *conversation initiation*, *giving and taking turns*, *giving and requesting feedback*, and *breaking away*. The same conversational behavior can contribute to the realization of different discourse functions and the same discourse function can be implemented using different combinations of conversational behaviors. For example, head nods can indicate agreement, or simply attention; and to indicate agreement a listener may nod or say "uh huh." The actual mapping between function and behaviors depends, among other things, on current availability of modalities and the current state of the conversation.

These functions can be accomplished because of the level of interpersonal *sychronicity*, *multithreadedness* and *entrainment*[1], in everyday conversation. In face-to-face conversation, there is synchronicity among behaviors such that the production of verbal and nonverbal behaviors is finely timed to occur together. Conversation is multithreaded, where different threads are carried out in parallel on different time-scales, ranging from the highly reactive ones occurring at a sub-second scale (such as feedback) to ones that span the whole conversation. Finally, entrainment refers to the fact that rhythmic movements in conversation become synchronized over time such that conversationalists are highly reactive to one another's conversational behaviors. It is for this reason that turn-taking is so fluent and yet so quick. Before even the end of an utterance, a listener has picked up on cues signaling that the end is near, and has begun to prepare her own turn-beginning. This entrainment is present at all of the threads of conversation, from picking up on feedback, to deciding what to talk about.

To further clarify the type of roles discourse functions serve, the contribution to the conversation can be divided into *propositional information* and *interactional information*. Propositional information corresponds to the content of the conversation and includes meaningful speech as well as gestures, facial expression, head movements and intonation used to complement or elaborate upon the speech content. Interactional information consists of cues that affect the conversational process and includes a range of nonverbal behaviors as well as regulatory speech such as "huh?" "Uh-huh".

In short, the interactional discourse functions are responsible for creating and maintaining an open channel of communication between the participants, while propositional functions shape the actual content.

Architectural Requirements

The construction of a computer character which can effectively participate in face-to-face conversation as described above requires a control architecture which has the following features:

- **Multi-Modal 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.

[1] Thanks to Livia Polanyi for pointing out the importance of entrainment in the current context.

- **Real-time** – We need to get beyond the "ping-pong" model of conversation which assumes the turn holder only outputs while the listener only inputs. The system must allow the speaker to watch for feedback and turn requests, while the listener 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 sub-second timescale. The architecture should reflect this fact by allowing different processes to concentrate on activities at different timescales.
- **Understanding and Synthesis of Propositional and Interactional Information** – Dealing with propositional information requires building a model of user's needs and knowledge. Thus the architecture must include both a static domain knowledge base and a dynamic discourse knowledge base. Presenting propositional information requires a planning module to plan how to present multi-sentence output and manage the order of presentation of interdependent facts. 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).
- **Conversational Function Model** – Explicitly representing conversational functions 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 translate input into functions, and functions into outputs. This also produces a symmetric architecture because the same functions and modalities are present in both input and output.
- **Modularity and Extensibility** – The architecture must be modular and extensible with respect to input and output modalities. Furthermore, this system is a growing testbed for theories of human face-to-face communication . Therefore, we will need to revise and extend it.

Related Work

Embodied conversational agents are a specific type of multimodal interface, so in presenting our architecture we must first review other conversational systems and multimodal interfaces in general.

One of the first multimodal systems was *Put-That-There*, developed by Bolt, Schmandt and their colleagues (Bolt 1980). *Put-That-There* used speech recognition and a six-degree-of-freedom space sensing device to gather user 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. 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 diectic gesture. When these words were encountered, the system analyzed the user's pointing gestures to resolve diectic references.

Koons continued this work by allowing users to maneuver objects around a two-dimensional map using spoken commands, deictic hand gestures, and eye gaze (Koons et al. 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, also depending on the linguistic input to guide the interpretation of the other modalities (Wahlster et al. 1991). Bolt and Herranz describe a system that allowed a user to manipulate graphics with two-handed semi-iconic gesture (Bolt and Herranz 1992). Using a cutoff point and time stamping, motions could be selected that related to the intended movement mentioned in speech. Sparrell used a scheme based on stop-motion analysis: whenever there was a significant stop or slowdown in the motion of the user's hand, then the preceding motion segment was analyzed for features such as finger posture and hand position (Sparell 1993).

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 non-verbal behavior as interactional information to control the pace of the user-computer interaction.

These limitations were partially overcome by Johnston et al. (Johnston et al. 1997), who described an approach to understanding of 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 could drive the recognition process. Multimodal input was represented in type-cast 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.

Missing from all these systems is a concept of input interpretation or output generation with respect to conversational function. That is, there is no conversational discourse structure applied over the multimodal input (no notion of "speaking turn" or "information structure" (Steedman 1991)). Therefore the role of gesture and voice input cannot be analyzed at more than a sentence-constituent replacement level.

Other researchers have built embodied multimodal interfaces that add dialogue and discourse knowledge to produce more natural conversational characters. For example, Peedy the parrot is an embodied agent that allows users to verbally command it to play different music tracks (Ball et al. 1997). Peedy integrates a simple conversational dialogue manager with spoken language input, reactive 3D animation, and recorded speech output. However, it only uses speech input, and so can not recognize low-level audio and visual cues and can only respond to the user at the end of every utterance. A more serious problem is that it maps semantic representation of input directly to a task-based representation ignoring conversational function entirely. This character's embodiment allows use of "wing gestures" (such as cupping a wing to one ear when the parrot has not understood a user's request) and facial displays (scrunched brows as the parrot finds an answer to a question). These behaviors, however, represent only behaviors, rather than instances of particular conversational functions. This means that particular behaviors cannot be generated as a function of what modalities are available, or what the previous behaviors have been, when a function is needed. In this system, too, there is no distinction between propositional and interactional behaviors such that both content and conversational process can be modeled.

Another example is Olga, an embodied humanoid agent that allows the user to employ speech, keyboard and mouse commands to engage in a conversation about microwave ovens (Beskow and McGlashan 1997). Olga has a distributed client-server architecture with separate modules for language processing, interaction management, direct manipulation interface output animation, all communicating through a central server. Olga is event driven, and so only responds to user input and is unable to initiate output on its own. In addition, Olga does not support non-speech audio or computer vision as input modalities.

Both Olga and Peedy use a linear architecture in which data flows from user input to agent output, passing through all the internal modules in between. Nagao and Takeuchi (Nagao and Takeuchi 1994) suggest a different approach. Their conversational agent is based on the subsumption architecture created by Rodney Brooks (Brooks 1986). In this case the agent is based on a horizontal decomposition of task-achieving behavior modules. The modules each compete with one another to see which behavior is active at a particular moment. Thus there is no global conversational state or model and the conversational interaction arises from the interplay between the different behavioral layers. Their agent responds to speech and gaze information, but coordination of the input analysis and output generation is also an emergent behavior, so precise control is impossible. The end result is that user input and agent output are decomposed according to task behaviors rather than conversational function.

Our current approach derives from our previous work on the Ymir architecture (Thórisson 1996). In this work the main emphasis was the development of a multi-layer multimodal architecture that could support fluid face-to-face dialogue between a human and graphical agent. The agent, Gandalf, was capable of discussing a graphical model of the solar system in an educational application. 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 backchannel behaviors that lead to a very natural conversational interaction. This work provided a good first example of how verbal and non-verbal 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. In contrast, "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 conversation between two artificial agents and the emphasis was on the production of non-verbal propositional behaviors that emphasized and reinforced the content of speech. The system did not run in real-time and since there was no interaction with a real user, the interactional information was very limited.

The approach we use combines lessons learned from both the *Gandalf* and *Animated Conversation* projects. In the next section we present a conversational function based architecture for developing embodied conversational interfaces. Following that we describe Rea, the first conversational humanoid based on this architecture.

Conversational Humanoid Architecture

Based on our previous experience with Animated Conversation and Ymir we have been developing an architecture that handles both real-time response to interactional cues and deep semantic understanding and generation of multimodal inputs and outputs[2]. At a high level (see Figure 2), our architecture is partitioned into: an Input Manager, which is responsible for collecting inputs across modalities; an Action Scheduler, responsible for synchronizing output actions across modalities; and components which handle the real-time interactional functions and longer-term deliberative responses such as content understanding and synthesis.

[2] This architecture has been developed in conjunction with the Conversational Characters project at Fuji-Xerox Palo Alto Laboratory.

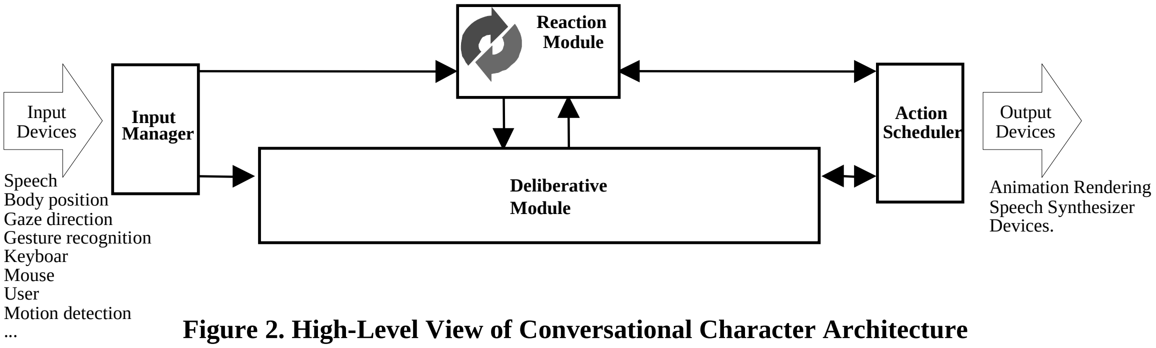

Figure 2. High-Level View of Conversational Character Architecture

In our implementation of this architecture, the modules communicate with each other using KQML (Finin and Fritzson 1994), a speech-act based inter-agent communication protocol, which serves to make the system very modular and extensible. Detailed characteristics of the modules in the architecture are described next.

Modules

Input Manager

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. Interactional information is also forwarded directly to the Reaction Module to minimize system response time.

The Input Manager will typically receive information from devices which 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. In our current implementation, the Input Manager also bundles co-temporal input events into aggregate semantic representations (e.g., a user utterance and accompanying gestures) for the Deliberative Module to process.

Reaction Module

The Reaction Module is responsible for the "action selection" component of the architecture, which determines what the character should be doing at each moment in time. The Reaction Module receives asynchronous updates from the Input Manager and Deliberative Module to determine the action to perform.

The Reaction Module currently responds to interactional cues based on the state diagram shown in Figure 3. The system starts up in the *NotPresent* state and remains there until a user is detected, at which time it transitions to *Present*. Once the user and the character have exchanged greetings (or other similar cues) the system transitions into turn-taking, depicted by the *UserTurn* and *ReaTurn* states. The *Conclude* state is used to handle user interruptions of the character, allowing it to continue to the end of a phrase boundary before giving the turn back to the user. The *Interrupt* state is entered if the system detects that the user has turned away. We anticipate adding more states as we begin to explore multi-sentential, mixed-initiative dialog.

Deliberative Module

The Deliberative Module performs detailed analyses of perceptual inputs, and planning and elaboration of action outputs. This processing is performed primarily to map signal features into conversational functions and back, and to process inputs and outputs at a semantic level.

Action Scheduling Module

The Action Scheduler is the "motor controller" for the character, 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.

Fulfillment of Architectural Requirements

We feel that the architecture described meets all of the requirements for an embodied conversational character that can participate in human face-to-face conversation. It is capable of reacting to and producing inputs and outputs across multiple modalities by mapping specific features of these modalities into conversational functions and using a uniform knowledge representation format (KQML) throughout the system. It can run in real-time, by providing immediate responses to interactional cues, and decoupling processes such as content understanding and synthesis, which can take seconds of response time. The architecture is able to work with both interactional and propositional information, in fact most KQML frames used within our implementation have slots for both kinds of input interpretations or output specifications present. One of the primary functions of the Deliberative Module in the architecture is to enable the separation of channel-specific features from conversational functions, allowing the Reaction Module to deal entirely at the functional level of abstraction. Finally, the use of a common KQML representation throughout, coupled with the disciplined use of functional descriptors allows the system to be very extensible with respect to input and output modalities, and modular with respect to plugging in new modules which implement alternative theories of discourse.

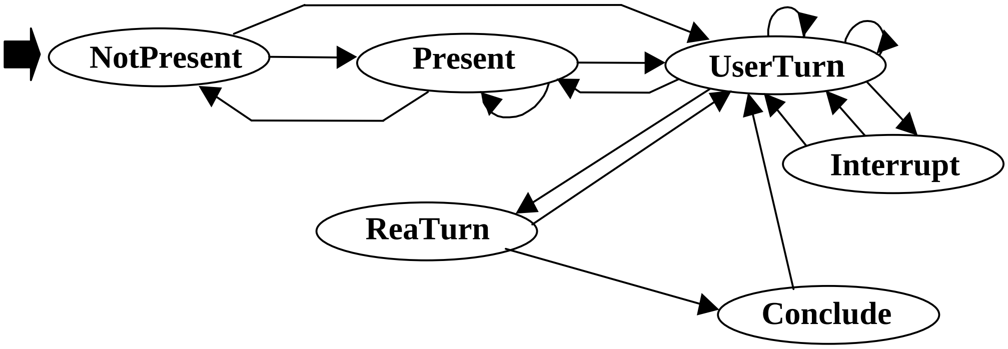

Figure 3. Conversational States used in Reaction

Implementation

Rea ("Real Estate Agent") is our first instantiation of the architecture described above. Rea is a computer generated humanoid that has a fully articulated graphical body, can sense the user passively through cameras and audio input, and is capable of speech with intonation, facial display, and gestural output. The system currently consists of a large projection screen on which Rea is displayed and which the user stands in front of. 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 (Figure 4).

The system is implemented in C++ and CLIPS, a rule-based expert system programming language (CLIPS 1994).

A Sample Interaction

Rea's domain of expertise is real estate and she acts as a real estate agent showing users the features of various models of houses that appear on-screen behind it. The following is a excerpt from a sample interaction:

Lee approaches the projection screen. Rea is currently turned side on and is idly gazing about. As the user moves within range of the cameras, Rea turns to face him and says "Hello, my name is Rea, what's your name?"
"Lee"

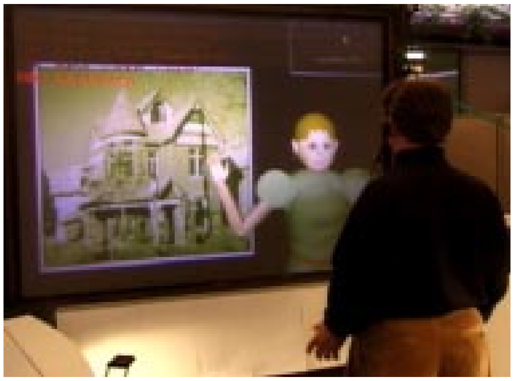

Figure 4. User Interacting with Rea

"Hello Lee would you like to see a house?" Rea says with rising intonation at the end of the question.
"That would be great"
A picture of a house appears on-screen behind Rea.
"This is a nice Victorian on a large lot" Rea says gesturing towards the house. "It has two bedrooms and a large kitchen ..."
Lee raises his hands into space, indicating his intention to take the turn, so Rea yields the turn to Lee.
"Tell me about the bedrooms," Lee says.
"The master bedroom is furnished with a four poster bed, while the smaller room could be used for a children's bedroom or guest room. Do you want to see the master bedroom?".
"Sure, show me the master bedroom". Lee says, overlapping with Rea.
"I'm sorry, I didn't quite catch that, can you please repeat what you said", Rea say.
And the house tour continues…

Rea is able to conduct mixed initiative conversation, describing the features of the house while also responding to the users' verbal and non-verbal input. When the user makes cues typically associated with turn taking behavior such as gesturing, Rea allows herself to be interrupted, and then takes the turn again when she is able. She is able to initiate conversational repair when she misunderstands what the user says, and can generate combined voice and gestural output. For the moment, Rea's responses are generated from an Eliza-like engine (Weizenbaum 1966), but efforts are currently underway to implement an incremental natural language generation engine, along the lines of (Cassell 1994).

In order to carry on natural conversation of this sort, Rea uses a conversational model that supports multimodal input and output as constituents of conversational functions. That is, input and output is interpreted and generated based on the discourse functions it serves. The implementation of the multimodal conversational model and the modules in the Rea architecture are discussed in the next sections.

Input Sensors

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 3D position and orientation of the head and hands at 10 to 15 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 are 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 reaction 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 Deliberative Module for interpretation.

Action Scheduler

The Action Scheduler coordinates output modalities through the use of event conditions specified on each output action which define when the action should be executed. For example, Figure 5 shows a set of typical commands sent to the Action Scheduler. The first is executed immediately (when: immediate) and causes the speech output channel to begin producing the string "I have just the house for you." The second action is only executed once the speech synthesizer begins actual production (potentially dozens of milliseconds after the command is issued) and causes the character's arms to be raised into a gesture-ready position. The third action is only invoked when the speech synthesizer begins production of the fourth word (index WORD3) and causes the character to point at an image of a house on the screen.

Each atomic output action is defined as a "behavior", which is a process capable of executing over some period of time. Behaviors can be invoked either as discrete ("point at", "beat gesture") or continuous ("walk forward", "gaze at user") actions. Behaviors can generate events when they start, terminate, and at other significant times (such as the speech channel's production of the *ith* word). These events form the basis of event-action conditions, which can be specified relative to arbitrary Boolean combinations of events and temporal constraints (e.g., "after speech has started, 10ms after the point gesture has ended, as long as it is before a specified absolute time"). Behaviors themselves register to receive events from their respective output devices, and respond to the callbacks by either signaling logical events (such as WORD3) or altering their execution (lip-synch is performed in this manner). We feel that such an event-driven Action Scheduler is necessary, since prediction of start times and durations for various channel actions is very difficult, if not impossible, especially in the distributed architecture we are developing.

Output Devices

The Action Scheduler drives the output devices, which include the computer graphics display and Rea's voice. It coordinates actions at the lowest level, such as eyebrow and eye movements, blinking, lip movements synchronized by phoneme callbacks, arm positioning for gestures, and body movements. We use Microsoft Speech Applications Programming Interface (SAPI) for the text to speech system that generates Rea's voice and intonation.

The graphical output of our conversational character is written in C++ using TGS's implementation of SGI's OpenInventor API with VRML extensions. Rea's body model is specified in VRML 2.0 and is based on the H-Anim VRML Humanoid Specification [17]. This currently includes a minimal set of body joints but a fully articulated set of arms and hands.

The animation engine performs keyframe animation with flexibility for custom interpolation functions. The internal representation of the body is hierarchical and divided into meaningful parts that are animated independently by the engine, allowing for easy asynchronous movement of independent body parts. The animation engine also takes responsibility for spontaneous generation of idle movements such as eye blinking to keep the body model in constant motion to enhance the believability of the character.

Conclusion

In this paper we have argued that embodied conversational agents are a logical and needed extension to the conversational metaphor of human – computer interaction. We argue that the nature of human face-to-face communication imposes strong requirements on the design of embodied conversational characters, and have described how our architecture satisfies these requirements.

We demonstrated our approach with the Rea system. Increasingly capable of making an intelligent content-oriented – or *propositional* – contribution to the conversation, Rea is also sensitive to the regulatory – or *interactional* -- function of verbal and non-verbal conversational behaviors, and is capable of producing regulatory behaviors to improve the interaction by helping the user remain aware of the state of the conversation. Rea is an embodied conversational agent who can hold up her end of the interaction.

```
[(action :id S :when immediate :cmd once
     :content (speak :content "I have just the house for you."))
 (action :when (on :event S.START) :cmd cont
     :content (gesture :cmd ready))
 (action :when (on :event S.WORD3) :cmd once
     :content (rightgesture :cmd point)) ]
```

Figure 5: Sample Action Scheduler Specification

Future Work

User-testing of Gandalf, capable of some of the conversational functions also described here, showed that users relied on the interactional competency of the system to negotiate turn-taking, and that they preferred such a system to another embodied character capable of only emotional expression. However, Gandalf did not handle repairs gracefully, and users hesitated comparatively more frequently when using the system (although they repeated themselves fewer times, and were more efficient in completing the task) (Cassell and Thórisson 1998). Our next step is to test Rea to see whether the current mixture of interactional and propositional conversational functions, including turn-taking and repair, allow users to engage in more efficient and fluent interaction with the system.

Implementing multimodal embodied conversational characters is a very complex undertaking, and we have an extensive research agenda of conversational competencies to add or improve on. High on our list is the implementation of a response planner which can synthesize natural language utterances and gestures based on conversational goals and context. We are in the process of integrating the SPUD system (Stone 1998) to perform this function. We are also currently exploring the implications of mixed-initiative, multi-sentential dialog on the architecture. Finally, the least reliable components of the system are the individual modality feature detectors, and we are continuing to refine and extend these to provide user gaze direction, facial expression, and gestural form.

Acknowledgements

Thanks to the other team members of the GNL group who have contributed to the development of Rea—Joey Chang, Sola Grantham, Erin Pänttäjä, and Jennifer Smith—and to the members of the FXPAL Conversational Characters project who have collaborated with us on the architectural design—Scott Prevost, Elizabeth Churchill, and Joe Sullivan.

References

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

Ball, G., Ling, D., Kurlander, D., Miller, D., Pugh, D., Skelly, T., Stankosky, A., Thiel, D., Van Dantzich, M. and T. Wax. Lifelike computer characters: the persona project at Microsoft Research. In Software Agents, J. M. Bradshaw (Ed.), MIT Press, Cambridge, MA, 1997.

Beskow, J. and McGlashan, S. Olga - A Conversational Agent with Gestures, In Proceedings of the IJCAI'97 workshop on Animated Interface Agents - Making them Intelligent, (Nagoya, Japan, August 1997), Morgan-Kaufmann Publishers, San Francisco, 1997.

Bolt, R.A. Put-that-there: voice and gesture at the graphics interface. Computer Graphics, 14(3), 1980, 262-270.

Bolt, R.A. and Herranz, E. Two-handed gesture in multi-modal natural dialog. In Proceedings of UIST `92, Fifth Annual Symposium on User Interface Software and Technology, (Monterey, CA, November 1992). ACM Press, 1992, 7-14.

Brooks, R.A. A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation. 2 (1), 1986, 14-23.

Cassell, J., Pelachaud, C., Badler, N.I., Steedman, M., Achorn, B., Beckett, T., Douville, B., Prevost, S. and Stone, M. Animated conversation: rule-based generation of facial display, gesture and spoken intonation for multiple conversational agents. Computer Graphics (SIGGRAPH ’94 Proceedings), 1994, 28(4): 413-420.

Cassell, J. and Thórisson, K. The Power of a Nod and a Glance: Envelope vs. Emotional Feedback in Animated Conversational Agents. Journal of Applied Artificial Intelligence, in press.

CLIPS Reference Manual Version 6.0. Technical Report, Number JSC-25012, Software Technology Branch, Lyndon B. Johnson Space Center, Houston, TX, 1994.

Finin, T., Fritzson, R. KQML as an Agent Communication Language. In The Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), ACM Press, November 1994.

Goodwin, C. Conversational Organization: Interaction Between Speakers and Hearers. New York, NY: Academic Press, 1981.

Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A. and Smith, I. Unification-based multimodal integration. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, Madrid, Spain, 1997.

Koons, D.B. Sparrell, C.J. and Thórisson, K.R. Integrating simultaneous input from speech, gaze and hand gestures. In Intelligent Multi-Media Interfaces M.T. Maybury (Ed.), AAAI Press/MIT Press, 1993.

Nagao, K. and Takeuchi, A. Social interaction: multimodal conversation with social agents. Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94), (Seattle, WA, August 1994), AAAI Press/MIT Press, 1994, vol. 1, 22-28.

Reeves, B. and Nass, C. The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places. Cambridge University Press, 1996.

Sparrell, C. J. Coverbal Iconic Gestures in Human-Computer Interaction. S.M. Thesis, MIT Media Arts and Sciences Section, 1993.

Specification for a Standard VRML HumanoidVersion 1.0. http://ece.uwaterloo.ca/~h-anim/spec.html

Steedman, M. Structure and intonation. Language, 1991, 67(2), 190-296.

Stone, M. Modality in Dialogue: Planning, Pragmatics, and Computation. PhD Thesis, University of Pennsylvania, 1998.

Thórisson, K. R. Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills. PhD Thesis, MIT Media Laboratory, 1996.

Wahlster, W., André, E., Graf, W. and Rist, T. Designing illustrated texts. In Proceedings of the 5th EACL (Berlin, Germany, April 1991), 1991, 8-14.

Weizenbaum, J. Eliza -a computer program for the study of natural language communication between man and machine. Communications of the ACM, 1966, 9, 26-45.