Social Performance Framework

Hannes Vilhjalmsson, Stacy C. Marsella
Center for Advanced Research in Technology for Education
USC Information Sciences Institute
Marina del Rey, CA
hannes@isi.edu, marsella@isi.edu

Abstract

The Social Performance Framework describes an approach to modularize the design of Embodied Conversational Agents. The focus is on the problem of generating verbal and nonverbal behavior that carries out an agent's communicative intent. The framework proposes modules and corresponding XML based interfaces that make a very clear distinction between intent and behavior. This division of labor between modules and well defined interfaces will allow ECA systems of different architectures to share important components that contribute to overall social believability.

Introduction

Research into Embodied Conversational Agents (ECAs) seeks to create artifacts that can engage us in social interactions. As the research has progressed, ECAs have become increasingly adept at having multi-modal interactions with humans that mirror human-to-human social interactions. And as they have become more adept, it has become feasible to envision and design systems where ECAs and humans can co-habit virtual and even real spaces to work, learn or play. However, this progress has typically required increasingly sophisticated architectures that address the many capabilities required to support multi-modal interaction: interpretation and generation of verbal and nonverbal behavior, ability to follow the norms that regulate social interaction, modeling of internal cognitive and emotional processes that underlie social interaction, the animation of the body, etc.

To address the complexity of ECA design and make progress in designing more sophisticated ECAs, the research community has relied on modular design principles. The decision of how to decompose an ECA system into modules has been driven by a number of influences: knowledge of human architecture, the modularity inherent in the available research on human behavior, constraints that flow out of available technology (such as speech generation systems) and not surprisingly the research interests of the researchers involved. Nevertheless, there is often considerable commonality between the various ECA designs.

In this paper, we discuss a framework to modularize ECA design and research. We call this framework the Social Performance Framework. The Social Performance Framework describes how an agent's intent to communicate can be brought from a conceptual level to an embodied action in a social setting. Instead of being a complete system architecture, it defines only certain key components and the interfaces between them. It does not make any assumptions about the type of implementation, which could range from a blackboard-like system to a messaging pipeline.

The Social Performance Framework derives much from previous approaches, while also proposing some new ideas on how to modularize ECAs. This is work in progress and will not only reflect our needs but also those of our collaborators. Some of the key design goals are as follows:

Manage complexity: Use a divide and conquer approach to divide the multimodal generation process into manageable system chunks, each with a clear role and well defined interfaces.

Support flexible intent to behavior mapping: There needs to be support for realizing the same communicative intent in multiple ways, based on information such as the physical state of the agent and the environment or different social/cultural settings.

Support variety of generation methods: No assumption can be made about how or in what order behaviors are generated. For example, we can not assume that the text to be spoken is available before gestures are generated. But what gets generated still needs to be linked such that semantic coherence is maintained.

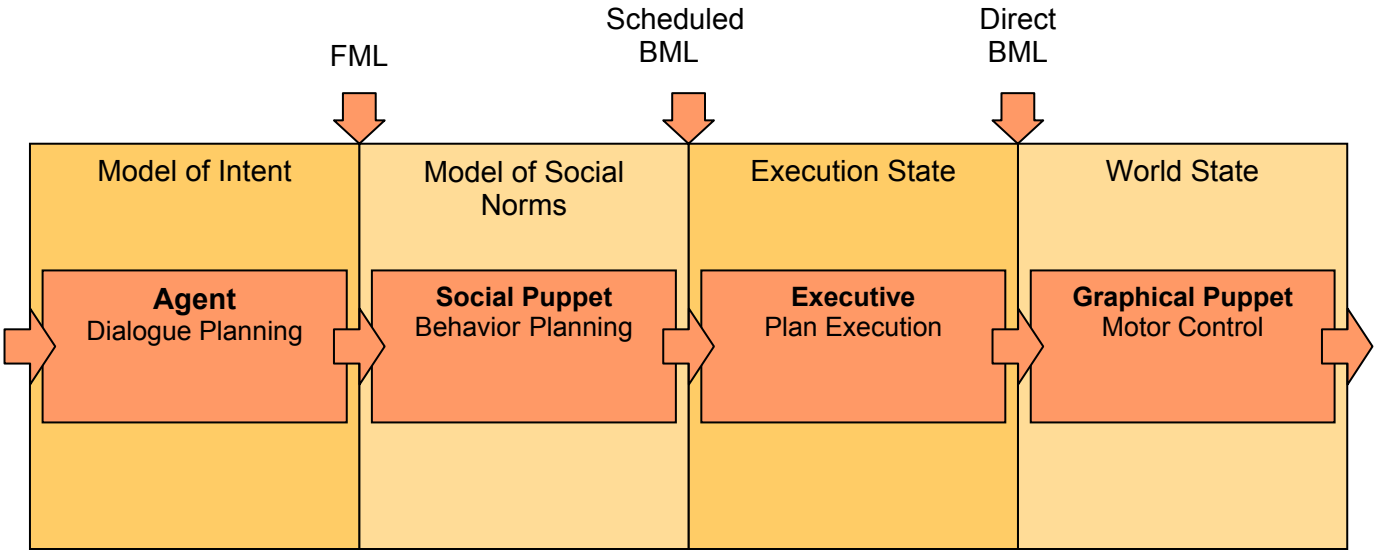

Figure 1: The Social Performance Framework

Support both Agents and Avatars: The graphical manifestation, or puppet, of an entity in a virtual environment may represent either an autonomous agent or a human's avatar. We propose that the puppets, whether driven by agent or human, share a part of the behavior generation framework because human users cannot be burdened with too much manual behavior control at a low level. Avatars need to exhibit enough autonomy to co-ordinate a nonverbal performance that reflects a user's communicative intent, but not more autonomy than that.

In light of these design goals, some key solutions we propose are as follows:

XML-based representation and interfaces: Not only does XML strike a good balance between flexibility and strict structuring, but it also a representation that is supported by a great number of tools and libraries, for almost any language.

FML vs. BML: A clear distinction is made between communicative intent or **function**, represented in XML as FML markup, and communicative **behavior**, represented as BML markup. These form interfaces at separate levels of abstraction.

Social Puppet: The agent's higher level intelligence is separated into a part that **models intent** and a part that appeals to a model of social norms to appropriately **carry out the intent** in a given setting. The former resides in a proper Agent module while the latter resides in a novel Social Puppet module. When driving avatars, human users would replace the top level agent module.

Related Work

The social performance framework is simply a new link in the long evolution chain of virtual humans and avatars. The emphasis on social competence, as manifested in appropriate nonverbal behavior, can be directly traced back to early Embodied Conversational Agents such as Gandalf (Thorisson, 2002) and STEVE (Rickel & Johnson, 1999). With Gandalf in 1995, Thorisson introduced a clear separation between dialog planning and social interaction control, arguing that what an agent chooses to say in a given situation is highly domain specific whereas the ability to deliver that content through social interaction is a broad re-usable skill (Thorisson, 2002).

This view was maintained in Gandalf's successor Rea (Cassell et al., 1999), that dedicated a special generation module to verbal and nonverbal behavior generation, taking an abstract level representation of communicative intent and giving it surface form according to the rules of social face-to-face interaction. Rea's generation module was a rather difficult piece to maintain and the interface not designed for general re-use.

The BEAT "text-to-embodied-speech" toolkit (Cassell et al., 2001), specifically addressed the re-use issue by introducing a plug-in model for nonverbal behavior generators and an XML-based processing pipeline. However, the generators had access to a variety of information about the text to be spoken, at different levels of abstraction, and therefore didn't quite provide a clean interface to communicative intent. The Spark system modified BEAT to work within an avatar-based chat system, using the behavior generators to automatically

animate the delivery of text messages between participants (Vilhjalmsson, 2005). The division between communicative intent and behavior was made very clear with Spark's first definition of FML and BML tag sets. The text messages were first annotated automatically with FML tags, describing communicative intent or functions, and then the generators would transform the FML tags into BML tags, turning communicative functions into the behaviors that would support them.

Another XML based behavior generation pipeline also embraced what has been called the bridging of mind and body, and proposed the Affective Presentation Markup Language or APML as an intermediary to express dialog moves at the meaning level, similar to FML (Carolis et al., 2004).

These markup languages were used to annotate the text to be spoken by a character, but were unable to express intent before text was made available or when speech was absent altogether. Another system, focusing on gesture planning, came up with the Multimodal Utterance Representation Markup Language or MURML, that specified cross-modal relationships within complex multimodal utterances in a more flexible manner than FML/BML or APML, using time markers instead of directly annotating the text (Kranstedt et al., 2002).

In order to coordinate these various representation efforts, several of the researchers mentioned here got together and came up with suggestions towards a common representation format at the 2005 "Representations for Multimodal Generation Workshop"[1]. The latest iteration of FML and BML presented in this paper follow these suggestions.

The Components

Figure 1 enumerates the basic components of the framework and labels the most important interfaces. Note this is not a system architecture, but simply an illustration of the key components in what we believe is a good framework. Other details such as data flow from right to left, shared memory between components and the ability to interrupt and re-plan are all architecture specific parts that will not be discussed further here. The rest of this section describes each of the components in turn.

Agent (Dialog Planning): This is the classic agent that perceives the world, reasons about it and determines an action to take. All of this happens at an abstract level and any communicative intent is expressed in the Communicative Functional markup Language or FML.

[1] See http://twiki.isi.edu/Public/MultimodalWorkshop

Social Puppet (Behavior Planning): Takes a description of communicative intent in FML and determines what steps are necessary to carry out that intent nonverbally in the current social setting according to a model of social norms. This generally breaks down into (a) *engagement* (making sure the agent is placed so that it can engage someone socially), (b) *interaction* (making sure protocols of interaction management such as turn-taking are followed) and (c) *proposition* (making sure verbal intent is effectively and appropriately communicated through co-verbal behavior). The output is a behavior plan that accomplishes these goals, described in a Communicative Behavior Markup Language (BML) wrapped inside an execution schedule.

Executive (Plan Execution). The Executive is responsible for executing the behavior schedule, sending individual behaviors to the graphical puppets as they come up in the schedule, and receiving notifications of their success or failure, possibly requiring some real-time choices to be made.

Graphical Puppet (Motor Control). The articulated figure inside the world that can perform behaviors when requested in BML, by handing them to various specialized motor controllers.

Each of these components has a well-specified role in the overall social performance of the agent and represent a fairly clean division of labor, both in terms of a system and in terms of research. Whether these components communicate through a blackboard or a pipeline is left up to the architecture, but the data that gets passed around needs to follow a certain interface specification.

The Interfaces

Communicative Function Markup Language (FML)

The Functional Markup Language specifies the communicative and expressive intent of the agent. A rich enough semantic description at this level then becomes the basis for both verbal and nonverbal behavior generation. An FML specification has two parts. The first part defines certain basic semantic units associated with the communicative event. The second part can further annotate these units with various communicative or expressive functions. The following is an example of an FML structure:

```
<participant id="ali" role="speaker"/>
<participant id="trainee" role="addressee"/>
<turn id="turn1" start="take" end="give">
  <topic id="topic1" type="new">
    <performative id="perform1" type="enquiry">
      <content>goal trainee ? here</content>
```

```
   </performative>
  </topic>
</turn>
<emphasis type="new">perform1:here</emphasis>
<affect type="fear">perform1:goal</social>
<social type="maintain_distance">trainee</social>
```

The units that are defined in this example include:

- **PARTICIPANT**: Names those that participate in this communicative event and identifies their roles that may include speaker, addressee, listener and over hearer.

- **TURN**: Indicates a single speaking turn. It also implies that the units within it can only be realized if the agent has received the turn. The attributes *start* and *end* further specify how the agent intends to bring that about (by requesting the turn or by taking it by force) and how the agent relinquishes the turn once done communicating (yielding the turn to everyone, giving it to the addressee or actually keeping it in case the agent wishes to continue speaking).

- **TOPIC**: Identifies a topic of discussion. All the units contained within this element belong to the same topic. The *type* attribute further specifies the type of topic shift, such as whether this is only a digression or a complete change to a new topic.

- **PERFORMATIVE**: Describes a core communicative goal in terms of a type and various CONTENT elements. The *type* could for example be *inform* or *enquiry*, based on whether the agent intends to share information with its addressee or ask a question.

- **CONTENT**: Each content element elaborates on the performative by describing the communicative goal in terms of relations between actions, people, objects and events. In this example, the goal of the trainee, in the location specified as *here*, is unknown. The ? marks an unknown value, which an enquiry will seek to fill in.

Any instantiation of these elements or anything contained within a CONTENT element becomes a semantic unit that can be further annotated with FML tags.

In this example, to emphasize the location when asking what the addressee is doing here, a new EMPHASIS element can be created around a reference to the *here* unit. Any unit can be referenced with some or all of its scope, as long as the reference is unambiguous. Here we could just as well use "turn1:topic1:perform1:here".

The initial set of FML elements that can operate on these semantic units includes:

- **AFFECT:** Specification of the affective state of the speaker, including whether this is the speaker's true emother emotion state or feigned and who/what is the target/cause of the emotion.

- **CONTRAST**: Speaker wants to contrast the enclosed unit with some earlier unit.

- **COPING**: Identification of a coping strategy employed by the speaker in relation to the contained unit.

- **SOCIAL:** Describes a social goal associated with the unit, such as to promote solidarity with a participant or increase social distance

- **COGNITIVE:** Describes meta-cognitive activity associated with the unit, such as possible difficulty in recalling it from memory.

- **CERTAINITY:** How assertive the speaker can be about a specified unit.

- **EMPHASIS**: Speaker wants listeners to pay particular attention to this part.

- **ILLUSTRATION**: Indicates a feature of a unit that could be clarified through illustration.

The FML language is very much a work in progress that will be influenced not only by our own needs but also the work and needs of other research groups that choose to share the framework (see pointer to workshop web page).

Communicative Behavior Markup Language (BML)

An agent's social puppet processes the communicative intent, once it has been described with FML, and suggests verbal and nonverbal behavior for carrying out that intent. These behaviors are described with the communicative behavior markup language or BML. The elements of BML roughly correspond to the body parts involved in the behavior and are further defined through sub-types and attributes. Every behavior is associated with a semantic unit, presumably the same semantic unit that gave rise to that behavior.

In our example, a process inside the social puppet may determine that the emphasis on location could be realized as a head nod and an eyebrow raise, and therefore suggest the following:

```
<head type="nod" amount="0.7" repeat="1">
perform1:here
</head>

<face type="eyebrows" side="both" amount="0.6"
shape="pointup" separation="0.5">
perform1:here
</face>
```

Furthermore, a natural language generation engine may have generated corresponding words:

```
<speech words="around here">
perform1:here
</speech>
```

This way a behavior schedule can guarantee that semantically linked verbal and nonverbal behavior occur together. This framework does not assume primacy of speech, so for example, a gesture that gets generated for a certain semantic unit may end up determining rate parameters for a speech synthesizer that synthesizes the words corresponding to that same unit.

As with FML, the BML language is work in progress and will reflect the kinds of behaviors we and other interested research groups are able to generate. The current set of BML elements includes:

- **HEAD**: Movement of the head independent of eyes. Sub-types include nodding, shaking, tossing and orienting to a given angle.
- **FACE:** Movement of facial muscles to form certain expressions. Sub-types include eyebrow movement, eyelid movement and expressive mouth shapes. It is also possible to specify an action unit according to the FACS model.
- **GAZE:** Coordinated movement of the eyes, neck, torso and legs, indicating where the character is looking.
- **BODY:** Full body movement, generally independent of the other behaviors. Sub-types include overall orientation, position and posture.
- **GESTURE**: Coordinated movement with arms and hands, including the sub-types point, reach, beat, depiction and signal. Also identifies phase of gesture (e.g., stroke, relax) to support more precise control over gestures.
- **SPEECH:** Concerned with verbal and paraverbal behavior. Sub-types include the words to be spoken (for example by a speech synthesizer), prosody information and special paralinguistic behaviors.
- **LIPS:** This element is reserved for controlling lip shapes by specifying visemes.

Each of these BML elements contains attributes that describe the visual appearance and movement dynamics of the behavior in order to achieve certain expressive effects. While BML provides a parameterized interface to procedurally generated behaviors, a less sophisticated system can also use these elements to index stored animation clips – excess parameters will then simply be ignored.

Scheduled and Direct BML

After the Social Puppet is done generating behaviors and aligning them with their semantic units, it constructs a BML schedule that specifies the order in which they need to be executed and any hard timing constraints that exist between them. The executive module then has the responsibility of starting execution, behavior by behavior. When it's time for a behavior to get executed, it is made available to the graphical puppet directly as a BML element. By monitoring the graphical puppet's progress with each behavior, the executive ensures the satisfaction of timing constraints.

Current Status

Two different projects have already started to adopt the general framework described here, with emphasis on the FML/BML interfaces. While their architectures are in fact different, we hope that they will be able to share some of the interesting pieces that contribute to a believable social performance. These projects are the Tactical Language Training System/BCBM project and the SASO/SmartBody project.

TLTS/BCBM

The Tactical Language Training System (TLTS), being developed at the Center for Advanced Technology in Education at USC/ISI, teaches basic survival skills in a foreign language and culture by combining intelligent tutoring and a 3D game environment (Johnson et al., 2004). In the game, students carry out missions in the foreign language that require them to build trust with local people by speaking with them and choosing the right gestures. These simulated social encounters and engaging story give the students a strong context for learning about the culture as well as practicing their new language skills.

The agents that drive the characters in the game at the level of intent are a part of a multi-agent system called PsychSim (Marsella et al., 2004). Each character also has a corresponding social puppet, an action executive and a graphical puppet inside the game world. Because there are many characters present, some coordination between characters is necessary, so all the social puppets belong to a social puppet manager that keeps track of who has the

current speaking turn and who has requested the turn. This manager may also allow a behavior generated by one social puppet to directly influence another social puppet to generate a corresponding behavior. A good example is a social puppet that wants to use a handshake behavior to realize the communicative intent to greet, requiring another agent's social puppet to reciprocate, resulting in an execution schedule governing two graphical puppets.

In TLTS, every dialog act chosen by an agent refers to line previously written by the game scriptwriters stored as an entry in an XML scene script. This line will also exist as a voice recording by a voice actor for that character. While the version of TLTS that is currently distributed for language training simply has the social puppet map from a chosen dialog act to a speech file and associated gesture animation, a next generation TLTS prototype has been implemented, as part of a separately funded project called BCBM, that makes full use of the FML/BML interfaces.

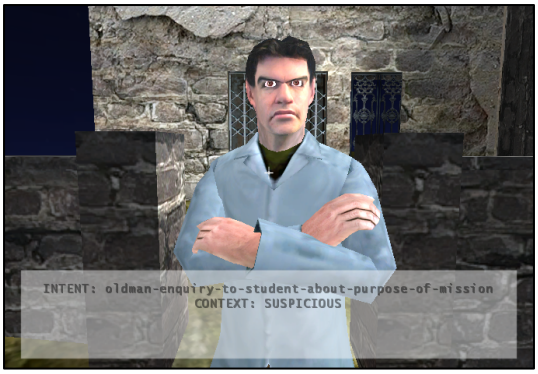

Figure 2a: The agent intent of enquiring about your mission with behaviors generated by the social puppet in the context of great suspicion.

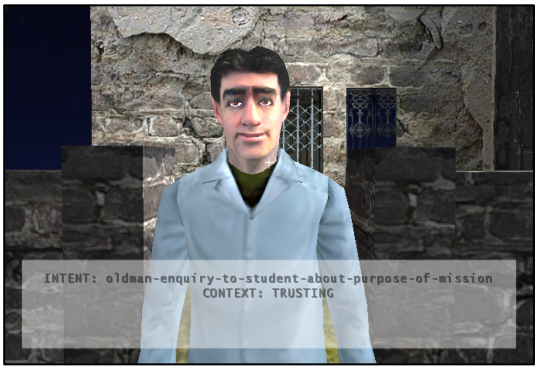

Figure 2b: The same agent intent, but now realized by the social puppet in a trusting context.

In this prototype, the scriptwriters add FML annotation to each line in the XML scene script using a specialized scriptwriting tool. An expert in the particular language and culture being modeled uses another tool to create general mappings between all the major FML elements to particular BML elements (Warwick and Vilhjalmsson, 2005). Each mapping choice can be made conditional on the value of special context variables that only are known at run-time. When the agents pick a dialog act, the whole FML annotated structure is then sent to the social puppet that consults the FML to BML map, using the current value of the context variables. Since the speech is pre-recorded, the behavior execution schedule ends up being tied to certain word timings that are made available to the social puppet module. An example of the same agent dialog act being realized by the social puppet in two different contexts is given in Figures 2a and 2b.

SASO/SmartBody

SmartBody is a new virtual human body being developed at University of Southern California (specifically two USC research institutes: the Information Sciences Institute (ISI) and the Institute for Creative Technology). Initially, SmartBody will become part of the SASO system, which grew out of the Mission Rehearsal Environment (Swartout et al., 2001). The SASO project is designed to teach leadership and negotiation skills under high stress conditions by immersing trainees in a realistic virtual world where they must negotiate with life-size virtual humans. Figure 3 is a scene from SASO depicting a Spanish doctor operating a clinic in Iraq. The trainee plays the role of an army officer that must convince the skeptical doctor to move his clinic out of harm's way.

The goal of the SASO's SmartBody project is to create a general purpose, highly capable and expressive, virtual human body that can be employed in a range of projects. In SASO, the SmartBody will be driven by a new version of the Steve agent (Rickel et al., 2002) that incorporates dialog management, the modeling of emotion and coping, natural language generation, social reasoning along with Steve's original planning and teamwork reasoning. This has several consequences for the use of the Performance Framework discussed above.

Figure 3: SASO's critical doctor

In particular, the communicative intent coming from the agent's "brain" is quite lush, including the speech acts,

semantic and pragmatic information, detailed information on the agent's emotional state, its coping strategy, and the syntactic structure of the surface utterance. This information can all be used to inform the behavioral performance and thus the FML has been designed to incorporate it. To support the role this detailed information can have on the performance, the BML has also been designed to support finer grain control, especially control over the phases of gestures and the facial expressions (based on the FACS model).

Conclusion

The Social Performance Framework is very much work in progress, but its adoption so far by two very different architectures and agent models, that will be able to share a model of social norms for behavior generation, speaks to the importance of the work and initial success. However, many challenges still remain. For example, we have not yet suggested a unified way to deal with arbitration between competing behavior generators. The framework also does not yet define how contextual information, such as discourse history and domain ontology are represented. Discussions about these issues as well as reports on the progress of our projects are forthcoming.

Acknowledgements

The authors would like to especially thank all the participants of the 2005 "Multimodal Representation Workshop" for many great ideas. Many thanks go to all the members of the TLTS/BCBM and SASO/SmartBody teams at ISI and ICT. The work presented here is made possible by several grants from the Defense Advanced Research Projects Agency (DARPA).

References

Carolis, B. D., Pelachaud, C., Poggi, I., & Steedman, M. 2004. APML, a Markup Language for Believable Behavior Generation. In H. Prendinger & M. Ishizuka (Eds.), *Life-Like Characters: Tools, Affective Functions, and Applications:* 65-86. Germany: Springer.

Cassell, J., Vilhjalmsson, H., & Bickmore, T. 2001. BEAT: the Behavior Expression Animation Toolkit. In *proceedings of SIGGRAPH'01*, Los Angeles, CA.

Cassell, J., Vilhjalmsson, H., Chang, K., Bickmore, T., Campbell, L., & Yan, H. 1999. Requirements for an Architecture for Embodied Conversational Characters. In N. Magnenat-Thalmann & D. Thalmann (Eds.), *Computer Animation and Simulation '99*. Vienna, Austria: Springer Verlag.

Johnson, W. L., Marsella, S., Vilhjalmsson, H. 2004. The DARWARS Tactical Language Training System. In *proceedings of The Interservice/Industry Training, Simulation and Education Conference*, Orlando, FL, December 2004

Kranstedt, A., Kopp, S., & Wachsmuth, I. 2002. MURML: A Multimodal Utterance Representation Markup Language for Conversational Agents. In *proceedings of AAMAS'02 Workshop Embodied conversational agents- let's specify and evaluate them!*, Bologna, Italy.

Marsella S., Pynadath D., Read S. 2004. PsychSim: Agent-based modeling of social interactions and influence. In *Proc. International Conference on Cognitive Modeling*, pp. 243-248.

Rickel, J. & Johnson, W. L. 1999. Animated Agents for Procedural Training in *Virtual Reality: Perception, Cognition, and Motor Control. Applied Artificial Intelligence*, 7(6): 523-546.

Rickel, R., Marsella, S., Gratch, J., Hill, R., Traum, D. and Swartout, B. "Towards a New Generation of Virtual Humans for Interactive Experiences," in *IEEE Intelligent Systems* July/August 2002, pp. 32-38

Swartout, W., Hill, R., Gratch, J., Johnson, W.L., Kyriakakis, C., Labore, K., Lindheim, R., Marsella, S., Miraglia, D., Moore, B., Morie, J., Rickel, J., Thiebaux, M., Tuch, L., Whitney, R. Toward the Holodeck: Integrating Graphics, Sound, Character and Story, in *Proceedings of 5th International Conference on Autonomous Agents,* Montreal, Canada, June 2001.

Thorisson, K. R. 2002. Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perception to Action. In D. H. Bjorn Granstrom, Inger Karlsson (Ed.), *Multimodality in Language and Speech Systems,* Vol. 19: 173-207. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Vilhjalmsson, H. 2005. Augmenting Online Conversation through Automatice Discourse Tagging. In *proceedings of HICSS 6th Annual Minitrack on Persistent Conversation,* Big Island, Hawaii.

Warwick, W. and Vilhjalmsson, H. 2005. Engendering Believable Communicative Behaviors in Synthetic Entities for Tactical Language Training: An Interim Report. In *Proceedings of the 14th Conference on Behavior Representation in Modeling and Simulation*, Universal City, CA, May 2005