Social Puppets: Towards Modular Social Animation for Agents and Avatars

Hannes Vilhjalmsson[1], Chirag Merchant[2], and Prasan Samtani[3]

[1] CADIA, Reykjavík University
Kringlan 1, IS-103 Reykajvík, Iceland
hannes@ru.is
[2] USC Institute for Creative Technologies
13274 Fiji Way, Marina del Rey, CA 90292, USA
merchant@ict.usc.edu
[3] USC Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292, USA
samtani@isi.edu

Abstract. State-of-the-art computer graphics can give autonomous agents a compelling appearance as animated virtual characters. Typically the agents are directly responsible for controlling their graphical representation, but this places too much burden on the agents that already deal with difficult high-level tasks such as dialog planning. This paper presents work, done in the context of an interactive language and culture training system, on a new kind of engine that fits between the high level cognitive agent models and the animated graphics that represent them. This is a social engine that generates socially appropriate nonverbal behavior based on rules reflecting social norms. Similar to modular physics engines, the social engine introduces a re-usable component that can heighten believability of animated agents in games and simulations with relatively little effort.

1 Introduction

Autonomous agents that interact with humans are found in applications ranging from health intervention to computer games. It is important for many of these applications to create a sense of face-to-face interaction with the agents and therefore they have benefited from modern graphics hardware that is capable of rendering a realistic physical appearance in real-time. After the agent software processes user input and generates agent responses, it typically calls a graphics engine to deliver speech and animation through an articulated face or body. This may suffice in a relatively constrained dialog environment, but take this into a dynamic 3D environment, such as the interactive world of games, and the physical delivery of spoken responses becomes more complex.

How does the animated body know that it is within hearing distance of its addressee before speaking? How does it visually indicate to those around it that it has something to say? How does it perform a specific co-verbal gesture when the spatial configuration of participants changes? How does it know it is not speaking out of turn? It is hard to avoid awkward social moments when the division between mind and body is absolute,

D. Schuler (Ed.): Online Communities and Social Comput., HCII 2007, LNCS 4564, pp. 192–201, 2007.

such as is the case when agent software, oblivious to physical surroundings, hands responses off to a graphics engine that is oblivious to the social situation.

It is possible to extend the original autonomous agent model to deal with all of these physical factors, but that places a lot of burden on a process that already has its hands full with coming up with the next thing to say. Besides, we should be able to generalize and re-use a model that carries out nonverbal behavior according to social norms. They are called norms for a reason.

In fact, this is similar to the situation where we have an agent that we want to behave realistically inside a world governed by Newtonian physics while also pursuing its high level goals. It would make for poor portability if we needed to re-implement our laws of physics every time we changed our agent models. Similar to attaching skeletal geometry to a "rag-doll" object inside a specialized physics engine and giving the physical simulation full control over its joints when the laws of physics need to apply, one can imaging plugging an agent into a social structure that ensures that the rules of social nonverbal behavior are observed as the agent pursues its goals in the world.

This paper describes work that was done as part of developing a system for rapidly teaching new languages and culture through an engaging social game environment. This overall system will be described in section 3 after the following review of related work. The role and implementation of the novel Social Puppets module will be discussed in section 4, followed by future work and conclusions.

2 Related Work

Agents simulating groups of people interacting with each other with or without a human in the loop, typically appear more believable when they act according to coherent social or psychological models, inspired by scientific theory and empirical data, than when they act in ad-hoc or random ways, even if their visual appearance is photo realistic [1], [13], [23]. This has encouraged researchers to build computational models and incorporate them into their agents' decision process. Implemented models include group dynamics [13], social role awareness [14], social relationship [2], politeness [24], emotion [7], [12] and personality, which tends to factor into many of these other models.

While all of the models mentioned address believability, they focus on the computation of the abstract inner state of agents and then how that state reveals itself through a choice of verbal action or perhaps facial expression. The nonverbal coordination of the social situation is often a secondary concern, which can leave these rich minds stranded in an awkwardly stiff or uncoordinated body.

Embodied Conversational Agents (ECA) [3] specifically address the nonverbal aspect of social conduct. This research generally draws from the study of human face-to-face conversation and applies rules that relate abstract description of communicative intent to observable physical behavior, which is realized through real-time multi-modal behavior production. Early ECAs, such as Gandalf [20] and Rea [4], demonstrated the importance of separating content generation and interaction control. It was argued that how and what an agent chooses to say in a given situation is highly domain specific whereas the ability to deliver the chosen content through face-to-face interaction with others is a broad skill and re-usable across domains.

Another important idea that came out of early ECA research was to keep the planning of communicative intent and planning of its surface form as separate stages in the production process. Wide adoption of this view and interest in sharing system components has lead to the formalization of a multi-modal behavior generation framework called SAIBA [9]. This framework defines an interface at the level of communicative intent, called Function Markup Language (FML) and another interface at the level of form description, called Behavior Markup Language (BML).

Primarily used as interface agents, Embodied Conversational Agents have mostly been built for one-on-one conversations with users in a relatively fixed physical setting. When moving into a dynamic 3D game environment, more behaviors and more complex patterns of interaction need to be considered, for example to deal with a larger numbers of participants and longer locomotion distances. Research into the generation of believable communicative behavior in multi-party settings is growing, but has for the most part focused on one or two kinds of behavior at a time such as posture or gaze [6], [15].

Another kind of research into the generation of multi-party social behavior deals with crowds, which has for a long time been at a level of detail that is too low for close quarters environments. However, recent work on autonomous pedestrians suggests the implementation of a coupling between cognitive control and reactive behavior control at the individual level to attain a higher level of realism in social locomotion [17] and work has started on simulating believable smaller sized crowds with a collection of rules based on statistical data on observed behavior in human gatherings [11]. While the detail in this work is not high enough to support face-to-face interaction, the gap that has existed between the deep modeling of a single individual in a very limited environment and the broader modeling of a large number of individuals in a complex environment is getting smaller. This trend is perhaps driven by the requirements of densely populated but highly interactive game worlds that are now possible.

It is important to build tools and flexible system frameworks to bring models of behavior into real-world applications and to speed up the development of new models and environments for testing. Such tools both exist for the more abstract socio-psychological models [16], [18] and for the rule-based generation of nonverbal behavior [5], [10]. The work presented here on Social Puppets, a special tool for game environments, is very much influenced by the latter, with roots in the Spark framework for animating online avatars [22] and its core engine which itself was based on the BEAT nonverbal behavior toolkit [5]. The Social Puppets approach aims to accommodate any kind of higher level agent models and lower level animation systems by supplying a clear behavior interface. The approach extends previous work by starting to address both the depth of face-to-face conversation and the existence of an extended 3D social game environment. The Social Puppets have been realized in the context of a real-world application which will be discussed next.

3 The Language and Culture Training System

The work presented in this paper was done as an important component of the DARPA funded Tactical Language and Culture Training System (TLCTS) which teaches

adults basic communication skills in a foreign language and culture [8]. The overall system combines several advanced technologies including speech recognition, dynamic learner modeling, adaptive feedback, interactive autonomous agents and a 3D game environment. Learners pick up new communication skills in a multimedia tutoring environment and get to practice them by switching to a game environment where they carry out related tasks within an interactive story. Advancing through the story relies on building trust with automated characters by speaking with them in their language and behaving in a culturally sensitive manner. Modules for Levantine Arabic, Iraqi Arabic and Pashto were developed at ISI and other languages and cultures are forthcoming from Alelo Inc., a spin-off that licensed the technology for commercialization.

The simulated social encounters in the game and the engaging story give learners a strong context for practicing the new language as well as learning about the culture. Nonverbal behaviors play an important role in any face-to-face interaction and are therefore a very important part of any language and culture training. From the inception of the project, it was clear that an accurate rendition of nonverbal behavior was essential.

A screenshot from the game in an early Pashto version of TLCTS is shown in Fig. 1. The learner, represented by an avatar (1), has just entered an Afghan village and is greeted by a group of children. Behind the learner stands a native guide who can assist if the learner stumbles (4). To interact with the children, the learner starts by crouching down, taking of his shades, to make eye-contact, and then greeting the children. The learner accomplishes the greeting by facing the children, selecting a hand-over-heart gesture with the mouse and speaking into a microphone (2). The way that the learner conducts himself affects the agents that control the characters of the children, possibly resulting in increased or decreased trust as indicated by an animated plus or a minus sign, and the movement of an accumulative trust bar underneath each character's portrait (3).

The TLCTS is a modular system with many well defined interfaces, several of which contribute to the game environment experience. For a description of all modules and the interaction between them, see [21]. The graphics are rendered in the Virtual Culture (VC) game engine from Alelo Inc., a modified version of the Unreal Engine from Epic Games. The VC engine has a character animation interface that supports procedural motor skills for communication, such as gaze control, facial expressions, pointing and detailed body locomotion and orientation.

Considerable work went into adding this repertoire of interpersonal behavior into the game engine because the original Unreal games were only concerned with combat related behavior. The game engine is also responsible for rendering and maintaining the graphical environment that serves as the stage for the interactive stories.

While the game engine carries out the final low level realization of character action, the decision about how a character responds to the learner's input, is taken at a much higher level in what is called the agent code. The agents, as well as the rest of the high-level processing in TLCTS, are written in Python and plug right into a flexible framework that supports message routing between components. In fact, two very different implementations of the agent code were built, and switching between them is quite trivial. The former implementation is a full-blown multi-agent system with deep social reasoning that includes theory of mind. Originally developed as a

Fig. 1. A scene from the Pashto version of TLCTS where the learner has encountered a group of children in an Afghan village

general social simulation tool called PsychSim, a special version called Thespian was created for TLCTS. Thespian addresses two important needs that arise in an interactive drama setting. The first being able to give the agents a pre-written story script as a guideline for their behavior through a process called fitting [19] and the second being able to enforce common social norms that govern conversation, including multi-party conversation [18]. The second agent implementation is based on finite state machines, and was created as a less computationally intensive alternative to the first one to improve agent response time at the cost of reduced reasoning power and dynamism. Both types of agent modules receive the learner input in the form of speech acts and return agent responses also in the form of speech acts.

The deliberation at the level of speech acts does not involve any detailed coordination of nonverbal behavior or in fact any interaction with the simulated physical environment. This is not necessarily a fundamental limitation of the underlying agent technologies, but has more to do with the fact that they were authored around mental models rather than physical ones. Not only does this leave a gap between the decisions that the agents make and their manifestation in the environment, but also begs the question what happens while the agents are not producing speech acts? This is where the Social Puppets come in. They ground the agents in their physical bodies within the environment as the next section will explain.

4 The Social Puppets

Instead of having the agent code interface directly with the game engine, each agent interfaces with a *Social Puppet* (see Fig. 2). While each puppet represents an individual, all the puppets belong to a single social environment overseen by a social engine or a manager that enforces social order. This social engine communicates with the game engine through an executive module that takes care of executing behavior scripts once they have been generated by the puppets.

To use Social Puppets, a system first needs to instantiate a Social Puppets Manager. Through the manager, individual Social Puppets are created and named, one for each agent and one for the human learner. The Manager keeps track of all the puppets, routes special communicative messages between them and dynamically organizes them into interaction groups. The manager needs to receive updates about learner input and a few perceptual updates from the game engine, but is otherwise self-contained with respect to generating appropriate reactive nonverbal behavior for all the puppets. The agent code can have as much control over its corresponding puppet as it wants, but generally it only interfaces with it when it wants to speak or when it wants to change a context parameter (see below).

While running, the manager turns learner input and character perceptual data, into meaningful communicative events described in an early version of the Function Markup Language (FML). These events are then routed to any puppets that are

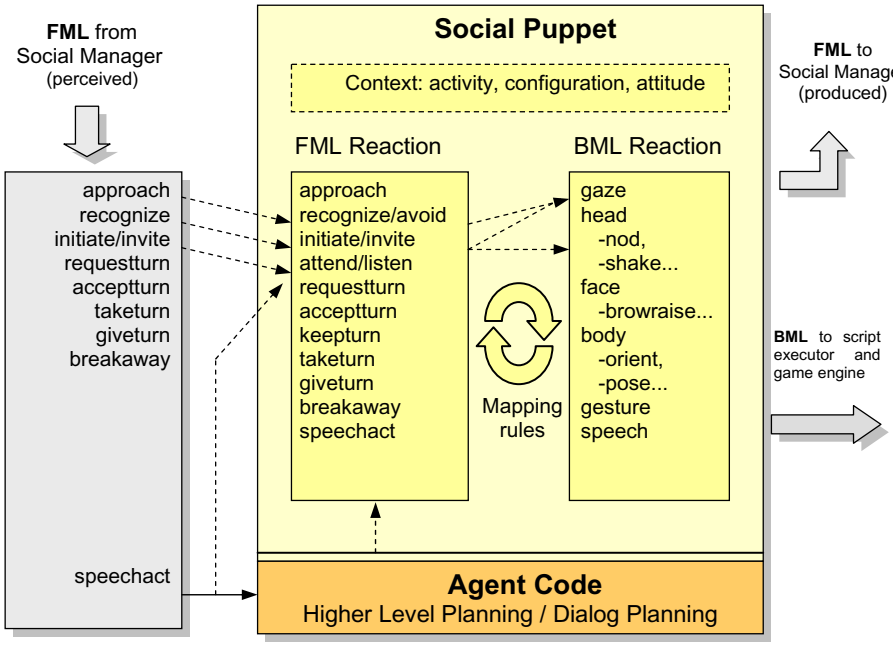

Fig. 2. The Social Puppet is coupled with an autonomous agent and takes care of adding non-verbal behavior, both in reaction to external communicative events and when the agent itself wishes to communicate (dashed arrows are example mappings)

possibly affected by the event, based mainly on how they are grouped. Everyone in a group gets to observe the same events, even if the event does not directly target them. For example, if the learner initiates contact with one member of a group, the other group members react.

When a puppet receives an FML event, it generates a communicative reaction, described at the same abstract FML level. The incoming event typically maps directly to a reaction such as accepting a conversation turn that has been given to the puppet. However, in some cases contextual parameters stored as a state vector in each puppet, have to be consulted. For example, a certain attitude can cause a puppet not to respond well to the approach of another puppet.

In the Pashto system, the learner might choose to approach an Afghan woman standing by herself near a well. As the learner approaches the woman, her puppet receives an approach message from the learner's puppet. The woman's puppet finds that its attitude parameter is negative and therefore selects avoidance as a reaction rather than recognition (see Fig. 3). The attitude value had been set as part of initializing the scene with proper cultural information. In this case it's part of Pashtun culture to condemn attempts from strangers, especially men, to interact with local women.

Once a puppet has chosen the appropriate reaction at the level of communicative intent, it now has to plan nonverbal behaviors that support this intent. There are several different ways to accomplish this. Mostly this is done as programmed procedures for each type of communicative event, and as a function of the contextual parameters. These parameters currently describe three dimensions of puppet state: Physical configuration (such as SITTING or STANDING), primary activity (such as READING or SOCIAIZING) and social attitude (such as HOSTILE or FRIENDLY). Other state parameters are possible, but so far this set has been found to be valuable in picking the most appropriate nonverbal behavior in TLCTS. The programmed procedures provide maximum flexibility for implementing relatively complex models of human social behavior.

For a more direct mapping, each puppet also keeps a four-dimensional behavior lookup table. The index into this table is the intent and the three current context values (any of which can be a "don't care" value) and what is returned is the best matching behavior description or animation name, as well as a new state vector and associated transition animation if needed (such as "standing up" if the puppet was in a "sitting" configuration).

The third way to generate behavior, and one that has not been fully exploited yet, is to use a file that contains FML to BML mapping rules built with a new visual application called "BCBM Rule Builder" [25]. These rules can tie together any FML representation with any contextual XML representation (defined by the author) to produce any nonverbal behavior performance described with a block of BML. The application allows the author to test the rules on a BML compliant character animation engine to explore in real-time the triggered nonverbal behavior, and therefore can greatly speed up the authoring process. A prototype with this functionality is already in Social Puppets.

Fig. 3. Examples of communicative intent turning into visible behavior in the Social Puppets. Avoidance (Left). Speaking and listening (Right).

After the puppet is done producing a behavior description in the form of a BML-level script, it is passed on to an execution module which in turn feeds the game engine with individual behavior commands that drive the character animation. The puppet's intent is also broadcast to any other relevant puppets through the manager to continue the sequence of events.

In some cases, the manager itself can choose to generate a sequence of events according to a particular behavior model. For example, the manager in Tactical Pashto implements the turn-taking model from [22] where it is assumed that the turn is returned to whomever spoke before the current speaker if no explicit turn action is taken. Because the manager keeps track of all puppets and their groups, it is a good place for implementing top-down behavior models for group behavior, whereas the puppets themselves are a better place for bottom-up rules that are meant to result in some emergent social order.

Speech acts are a special kind of communicative event in the social engine. These either come from the learner or the agent code as mentioned above. The pathway for these events is different from other communicative events. Speech acts from the learner trigger turn-taking events in the social puppets, including the puppet that represents the learner, but the acts are also routed to the agents so they can generate a response. The response from an agent is a speech act that gets passed through the agent's puppet, generating nonverbal behaviors, before finally coming out in the environment as a multi-modal performance.

5 Conclusions and Future Work

The Social Puppets have not been formally evaluated as a component by themselves, but the rigorous testing and subsequent release of the overall TLCTS to thousands of end users speaks well of the module's robustness, and the warm reception, of its quality of output. Furthermore, the module visibly improved development time, not least because it provided a new middle-level for scripting prototype characters that

didn't require any agent code to be present. Currently, one of the biggest problems with the approach is that while channels for communicative events and behaviors are well defined, other required information, such as perceptual data and contextual parameters, hasn't lent itself to clear-cut modularization and therefore some ad-hoc connections still remain. Future work will involve cleaning this up, extending the range of communicative intent and behaviors, and including dynamic locomotion planning using social anchor points such as formations and environmental features.

Acknowledgements. This project was a part of the DARWARS Training Superiority Program of the Defense Advanced Research Projects Agency and was conducted while the first author was working at USC/ISI. The authors wish to acknowledge the contributions of the members of the Tactical Language team and the BCBM team at Micro Analysis and Design. Many thanks to W. Lewis Johnson, Andre Valente and Stacy Marsella for their leadership and support.

References

1. Bailenson, J., Blascovich, J.: Avatars. In: Bainbridge, W.S. (ed.) Encyclopedia of Human-Computer Interaction. Berkshire Publishing Group, pp. 64–68 (2004)
2. Cassell, J., Bickmore, T.: Negotiated Collusion: Modeling Social Language and its Relationship Effects in Intelligent Agents. In: User Modeling and User-Adapted Interaction, vol. 13, pp. 89–132. Kluwer Academic Publishers, Boston (2003)
3. Cassell, J., Sullivan, J., Prevost, S., et al. (eds.): Embodied conversational agents. MIT Press, Cambridge (2000)
4. Cassell, J., Bickmore, T., Billinghurst, M., et al.: Embodiment in Conversational Interfaces: Rea. CHI, pp. 520–527. ACM Press, New York (1999)
5. Cassell, J., Vilhjalmsson, H., Bickmore, T.: BEAT: The Behavior Expression Animation Toolkit. SIGGRAPH, pp. 477–486. ACM Press, New York (2001)
6. Gillies, M., Ballin, D.: A Model of Interpersonal Attitude and Posture Generation. In: Intelligent Virtual Agents, Springer-Verlag, Heidelberg (2003)
7. Gratch, J., Stacy, M.: Evaluating the Modeling and use of Emotion in Virtual Humans. In: Autonomous Agents and Multi-Agent Systems, ACM Press, New York (2004)
8. Johnson, W.L., Marsella, S., Vilhjalmsson, H.: The DARWARS Tactical Language Training System. The Interservice/Industry Training, Simulation and Education Conference, SSA (2004)
9. Kopp, S., Krenn, B., Marsella, S., et al.: Towards a Common Framework for Multimodal Generation in ECAs: The Behavior Markup Language. In: Gratch, J., Young, M., Aylett, R., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, Springer, Heidelberg (2006)
10. Lee, J., Marsella, S.: Nonverbal Behavior Generator for Embodied Conversational Agents. In: Gratch, J., Young, M., Aylett, R., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 243–255. Springer, Heidelberg (2006)
11. Patel, J., Parker, R., Traum, D.: Simulation of Small Group Discussions for Middle Level of Detail Crowds. Army Science Conference (2004)
12. Pelachaud, C., Bilvi, M.: Computational model of believable conversational agents. In: Huget, M. (ed.) Communication in MAS: Background, Current Trends and Future, Springer-Verlag, Heidelberg (2003)

13. Prada, R., Paiva, A.: Synthetic Group Dynamics in Entertainment Scenarios. In: International Conference on Advances in Computer Entertainment Technology, ACM Press, New York (2005)
14. Prendinger, H., Ishizuka, M.: Social Role Awareness in Animated Agents, AGENTS'01. ACM Press, New York (2001)
15. Rehm, M., Andre, E., Nisch, M.: Let's Come Together - Social Navigation Behaviors of Virtual and Real Humans. In: Maybury, M., Stock, O., Wahlster, W. (eds.) INTETAIN 2005. LNCS (LNAI), vol. 3814, pp. 124–133. Springer, Heidelberg (2005)
16. Rehm, M., Endrass, B., Andre, E.: A Plug-and-Play Framework for Theories of Social Group Dynamics. In: Gratch, J., Young, M., Aylett, R., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 465–466. Springer, Heidelberg (2006)
17. Shao, W., Terzopoulos, D.: Autonomous Pedestrians. In: ACM SIGGRAPH Symposium on Computer Animation, ACM Publishing, New York (2005)
18. Si, M., Marsella, S., Pynadath, D.: Thespian: Modeling Socially Normative Behavior in a Decision-Theoretic Framework. In: Gratch, J., Young, M., Aylett, R., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 369–382. Springer, Heidelberg (2006)
19. Si, M., Stacy, M., Pynadath, D.: Thespian: Using Multi-Agent Fitting to Craft Interactive Drama. In: International Conference on Autonomous Agents and Multi-Agent Systems, pp. 21–28. ACM Press, New York (2005)
20. Thorisson, K.R.: Real-Time Decision Making in Multimodal Face-to-Face Communication. In: Autonomous Agents, pp. 16–23. ACM Press, New York (1998)
21. Vilhjalmsson, H., Samtani, P.: MissionEngine: Multi-System Integration using Python in the Tactical Language Project. PyCon, Python Software Foundation (2005)
22. Vilhjalmsson, H.: Animating Conversation in Online Games. In: Rauterberg, M. (ed.) ICEC 2004. LNCS, vol. 3166, pp. 139–150. Springer, Heidelberg (2004)
23. Vinayagamoorthy, V., Gillies, M., Steed, A., et al.: Building Expression into Virtual Characters. EUROGRAPHICS State of The Art Report, vol. 2006. The Eurographics Association (2006)
24. Wang, N., Johnson, W.L., Mayer, R.E., et al.: The Politeness Effect: Pedagogical Agents and Learning Outcomes. In: International Journal of Human-Computer Interaction, vol. 22, Lawrence Erlbaum Associates, Mahwah (2007)
25. Warwick, W., Vilhjalmsson, H.: Engendering Believable Communicative Behaviors in Synthetic Entities for Tactical Language Training: An Interim Report. Behavior Representation in Modeling and Simulation, SISO (2005)