Engendering Believable Communicative Behaviors in Synthetic Entities for Tactical Language Training: An Interim Report

Walter Warwick, Ph.D.
Micro Analysis and Design
4949 Pearl East Circle, Suite 300
Boulder, CO 80301
303-442-6947
wwarwick@maad.com

Hannes Vilhjalmsson, Ph.D.
Center for Advanced Research in Technology for Education
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA, 90292
310-448-8210
hannes@isi.edu

Keywords:
Communicative Behaviors, Gesture, Tactical Language Training, Middleware Development, Unreal Tournament

ABSTRACT: *In this paper we describe an ongoing research and development effort that supports the representation of coordinated verbal and nonverbal behaviors in synthetic characters. The ultimate goal of this effort is to allow a user to populate a virtual environment with characters that exhibit believable communicative competence. In this way, we will make computerized language training more engaging and more effective. Our work is focused on developing a middleware that supports two levels of abstraction. The first allows the user to specify a character's communicative intent and the relevant contextual attributes that underlie dialog in the synthetic environment. The second allows the user to specify how those intents might be variously realized in the virtual environment at run time. Although our efforts are geared toward improving the realism of a particular simulation-based training system, this work also demonstrates the reciprocal ability of modeling and simulation to advance theory; insofar as the development of a high-level specification constrains the behavior of synthetic entities, it also lays bare theoretical assumptions about what we take to be the salient aspects of those behaviors and puts them to the test in a synthetic environment.*

1. Introduction

A tactical language comprises a set of communicative acts, including the verbal and nonverbal behavior necessary to accomplish specific tactical missions. In the absence of the verbal fluency that can take years to achieve, nonverbal cues take on an even more important role in the communicative process. These cues exist in a variety of functional forms, all of which are integral to achieving the communication goal in a tactical language encounter. Furthermore, cues in any form that are misplaced, incorrect, or simply missing can lead to negative training and serious repercussions for forces on the ground.

The most consciously produced are emblematic and propositional gestures, such as "thumbs up" indicating positive feedback in American culture, or pointing to indicate "move that over there" (Cassell, 2000) Both types are used as clearly identifiable signs that denote a specific meaning. Emblematic gestures in particular can be considered as physical "words" in a language; as such, they are culturally specific and as susceptible to misinterpretation as an idiomatic phrase. The third and most frequent form is the spontaneous gesture, which occurs with natural speech flow. Because spontaneous gestures are not usually conscious to either speaker or listener, they are also those which are most notably lacking in the design of synthetic agents. However, since these gestures act as central "vehicles for our communicative intent," excluding them from an agent's behavioral repertoire results in a greater degree of misunderstanding than even the incorrect use of emblematic gestures (Cassell, 2000). The "simple" acts of approaching another human, initiating an interaction, and maintaining it in an orderly fashion, are crucially

dependent on the precise display and timing of these subtle behaviors.

The University of Southern California's (USC) Center for Advanced Research in Technology for Education/ Information Sciences Institute, with support from Micro Analysis and Design, is currently developing the Tactical Language Trainer (TLT), a computer-based training system for the rapid acquisition of mission-oriented communication skills, both verbal and non-verbal (Johnson, et al., 2004). A central component of the TLT is a Mission Practice Environment that allows the learner to practice communication skills with synthetic characters in a variety of mission scenarios. Currently, significant effort is required to produce coordinated verbal and nonverbal behavior in the synthetic characters that populate the mission scenarios. Moreover, the process of coordinating those behaviors is undertaken at a programming level within the Mission Environment and is thus generally beyond the reach of the scenario developer.

In this paper we describe an ongoing project to develop a middleware layer that will support the rapid and efficient specification of communicative intent and with it the coordination of verbal and nonverbal behaviors at a high-level. We begin with a simple, but concrete example of a training scenario within the Mission Environment and we indicate the role our middleware plays in the development of such an example. Next, we discuss two levels of abstraction supported by our middleware. The first allows scenario developers to specify a character's communicative intent and the relevant contextual attributes that underlie dialog in the synthetic environment. The second allows the scenario developer to specify how those intents might be variously realized in the virtual environment at run time. Then, we describe some of the more general implications we see in this work for modeling and simulating human behavior. Although our efforts are geared toward improving the realism of a particular simulation-based training system, we find ourselves essentially developing a high-level specification to constrain the behavior of synthetic entities. The decisions about which behaviors to include and how to specify them turn on interesting questions about the invariant features of the domain being modeled and the fidelity of the simulation used to model the domain. Finally, we conclude with a brief outline of the work that remains to be done on the project.

2. An Example of Tactical Language Training in the Mission Environment

As we indicated above, the Mission Environment of the TLT allows a student to interact with synthetic characters in a computer-based training environment. The student is represented by an avatar in the synthetic environment. As other characters speak to his avatar, the student must listen to and understand their utterances and then respond (via a microphone) with the correct verbal responses while directing (via keyboard and mouse) his avatar to produce non-verbal behaviors appropriate to the situation. The reaction of the synthetic characters and the direction in which the scenario unfolds depend on the student's response.

To use an example scenario already developed by USC, the student must enter a cafe and ask for directions. Along the way, the student must demonstrate a working knowledge of cultural norms (e.g., by correctly identifying whom to speak to), he must produce utterances and gestures appropriate to the situation (e.g., manifesting either deferential requests and gestures to defuse a tense situation or more assertive commands and gestures to cajole a recalcitrant character) and, finally, he must comprehend and follow the directions given to him. The synthetic environment is depicted in Figure 2.1 below.

Figure 2.1. The cafe scenario in the Mission Environment. In this example, the student, represented by an avatar (second character from the left), must build trust with and receive information from the older character at the table.

Although the synthetic characters are capable of responding dynamically to the student input, the coordination of their verbal and nonverbal behavior (e.g., the deferential movement of the character's hand across his chest as he speaks) is still largely scripted at a low level of detail; the association of deference with the utterance occurs deep in the "Mission Engine" software that underlies the Mission Environment, far from the level where the scenario developer can reach it. Moreover, the rendering of deferential intent in the synthetic environment as this gesture rather than another is also far removed from the scenario developer. The middleware we're developing exposes these associations as mappings between high-level abstractions that can be easily modified by a scenario developer or an expert on language and nonverbal behavior rather than a software developer.

Thus, given the utterances of each character, a scenario developer is able to annotate the utterances with "tags" corresponding to communicative intents. Moreover, the mapping between these tags and the manner in which they are realized is also exposed to the scenario developer. So, in the example above, the scenario developer would be able to indicate that the synthetic character will engage the student deferentially, and further, that differential intent is realized with a particular shift in posture.

3. A Middleware to Support Two Levels of Abstraction

The middleware we're developing bridges a gap between scenario development and scenario execution in the TLT. This bridge comprises two graphical user interfaces. The first augments an existing interface used to specify dialogue in the Mission Environment. The second interface is new to the TLT. We describe each interface below.

3.1 The Script Writer

The first step in building a scenario is to create a branching dialogue script that describes a typical interaction between the student and the characters in the scene. This script provides a set of available dialogue moves for the autonomous agents that drive each character. At run-time, the agents pick moves in response to student actions, according to the rules of a drama engine called Thespian, which is based on the multi-agent PsychSim platform (Marsella and Pynadath, 2004). While the actual utterances that the synthetic characters can speak are pre-stored at this point, the accompanying nonverbal behavior does not have to be. Rather, after specifying the dialogue moves for each character, the scenario developer will annotate each utterance of the script with the communicative intent of that character. As we discuss in more detail below, these annotations represent high-level intentions—e.g., *engage conversation, elicit an answer, give/take-turn*, *emphasize point*, etc. Apart from some very basic timing information specific to the utterance structure, the annotations are quite general. The idea here is not to force the user to map specific utterances to specific gestures, but rather allow the user to indicate what a speaker hopes to accomplish, which can later be reconciled against contextual factors to produce a specific series of supporting gestures. Once specified, these annotations are first saved as an XML formatted file and then ultimately passed to the Mission Engine at run-time so that the appropriate nonverbal behaviors can be determined and synchronized with the character's speech.

To realize this functionality, we are extending the "Script Writer" interface of the TLT (depicted in Figure 3.1.1 below). In addition to specifying the utterances themselves, the user can choose from a list of intents to be associated with each utterance. Time markers are also made available to anchor intent to specific points in the utterance. Finally, the interface will ensure that place holders used in the high-level specification of intents are bound to the appropriate tokens in the particular scene being defined by the scenario developer. Returning to the cafe example, the scenario developer might annotate the utterance, "What are you doing in this region?" (spoken in Iraqi), as being of a complex intent type "enquiry." This, in turn, requires that referents for "you" and "this region" are defined so that whatever deictic gestures that happen

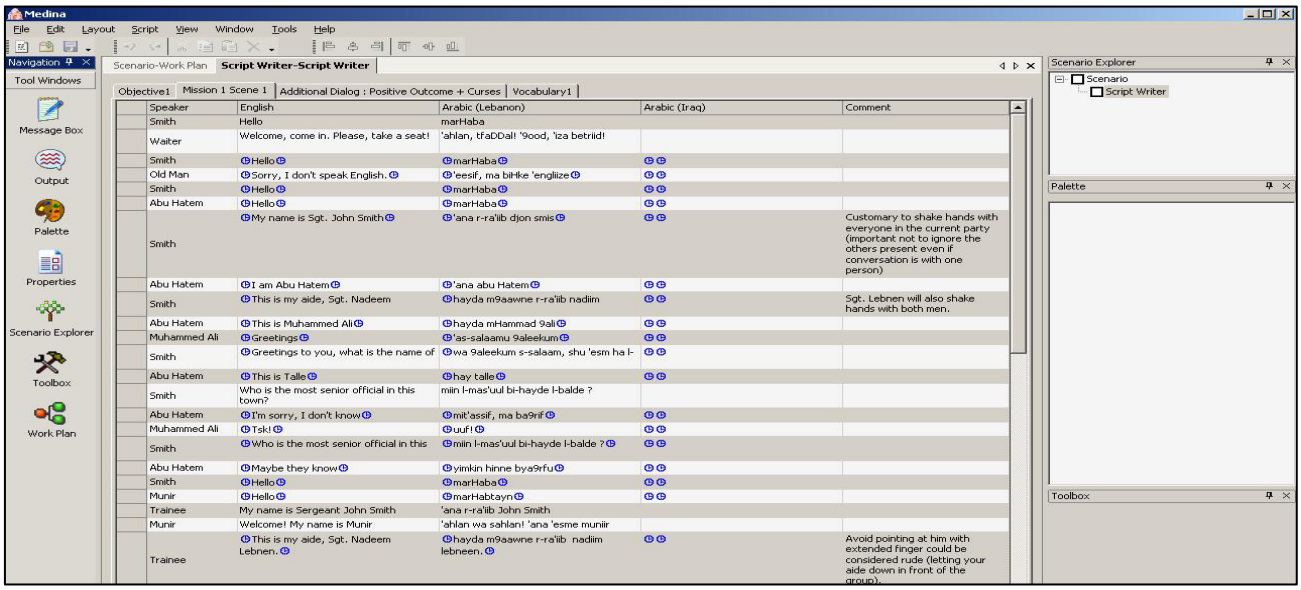

Figure 3.1.1. The Script Writer interface that we are extending to support the specification of communicative intent.

to constitute the intent are targeted in a plausible fashion—namely toward the student's avatar for "you" and a more general indication of "here".

3.2 Mapping Intent to Behavior

Depending on the context, the communicative intent of an enquiry might be realized in different ways. For example, a trusting speaker might maintain a closer proximity to his interlocutor than a suspicious speaker would. Likewise, even given the same utterance, the two speakers are likely to place a different level of emphasis on the same words, employ different gestures and maintain different levels of eye contact. Thus, the scenario developer might wish to define two realizations for the same intent, each of which reflects a different context. The scenario developer might even want to see how different gestures affect realism for a single intent type in a single context.

In earlier versions of the TLT, the mapping between intents and nonverbal behaviors was hard coded deep in the TLT software. We have developed a new interface that allows the scenario developer, or someone who has specialized knowledge about nonverbal communication, to create this mapping without having to modify source code in the TLT. Instead, the interface allows the scenario developer to select an intent type (e.g., "enquiry"), match it to a context (e.g., "trusting") and then define the behaviors that will constitute the intent (e.g., gaze, posture, intensity and size of gesture, etc.). Note that the behaviors the scenario developer selects are still specified well above the level of description needed to render a character in the synthetic environment; that is, there is still yet another mapping that must be made between behaviors and specific animation commands. But that is as it should be. The scenario developer is interested in teaching the student to recognize certain intents in different contexts and not, for example, what degrees of freedom in the articulated virtual human need to be acquired to achieve a realistic head tilt or what joint angles need to be set to point at a target.

The mapping is accomplished by way of the interface depicted in Figure 3.2.1. As indicated above, certain aspects of the behavior can only be defined in general terms that are made specific when the intent is applied by way of the Script Writer to a particular scene. Thus, timing information is specified in relative terms. Once the user defines the relationship between intents, contexts and behaviors, the mapping is saved as an XML-formatted file that can be read by the Mission Engine.

4. Implementing and Testing a Taxonomy for Nonverbal Behaviors in a Synthetic Environment

We have followed a spiral development process in our middleware construction. For this reason, the decision as to what communicative intents to include in the first spiral has been largely pragmatic; this set had to be small enough to be manageable, but large enough to demonstrate the potential impact of nonverbal behavior on the synthetic character's training effectiveness. In general, however, the decision as to which intents and behaviors to include in the middleware rests on more theoretical concerns having to do with how various intentions might be neatly aggregated and parameterized under more general communicative functions and contextual attributes. The set of intents we ultimately define will represent a taxonomy which should, in principle, reflect the "natural kinds" of communicative intent. Hitting the right kinds is important, especially if the character's communicative intent is to drive other aspects of his cognitive activity (e.g., a speaker who intends to accept blame might not only change his manner of nonverbal communication, he might also change his tactical plans). If the taxonomy is too sparse we risk not being able to individuate nonverbal (and other) behaviors; if the taxonomy is too dense, we risk losing the utility of abstraction by overcomplicating the functional relationship between intents, contexts and behaviors.

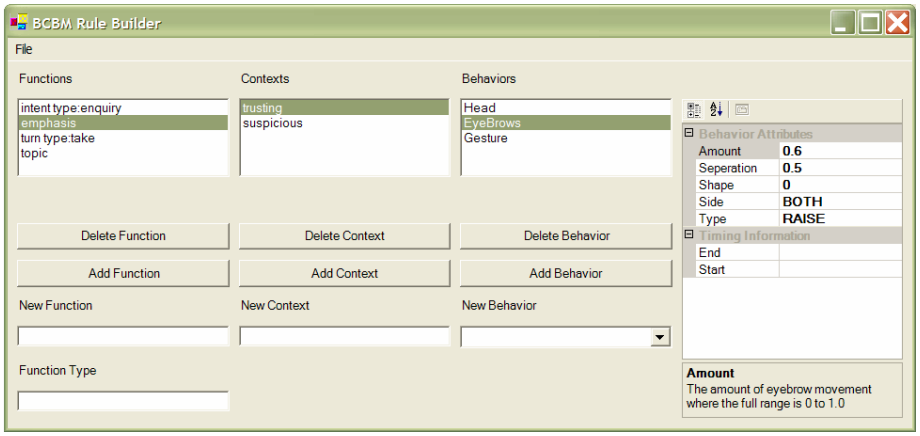

Figure 3.2.1 Communicative intent, in a certain context, results in certain behavior. This mapping can be edited through the Rule Builder interface.

The question, then, is how to determine when we've hit upon the right taxonomy. Existing research and observation suggest that for a given communicative intent different people in different situations will produce a wide variety of nonverbal performances. For example, one person may realize "get attention of X" by calmly approaching X and raising the chin and eyebrows when visual contact is made, while another person may frantically shout

and wave from a distance. Moreover, research to date that has classified communicative intent and described typical supporting nonverbal behavior typically does so without fully explaining the effects of a range of contextual factors, or it singles out certain core factors, such as discourse context (Cassell, et al., 2001) or emotion (Marsella and Gratch, 2003). However, no single model and authoring framework combines communicative intent with the number of contextual factors and resulting behaviors that we could represent within the TLT.

The opportunity exists here to use the middleware we are developing to produce and evaluate just such a model. That is, by allowing a researcher first to map intents to contexts to behaviors and then to "play" those behaviors to a student whose job it is to recognize the character's intent, we give the researcher a tool to experiment with different mappings. A mapping that plays especially well to students suggests that the researcher might be getting closer to representing and, hence, identifying the right kinds of relationships between context and communicative intent. Moreover, a computational representation yields a very precise and quantitative description of the relationships in question. Thus, we might arrive an account of a suspicious enquiry literally broken down in terms of its component proportions of, say, gesture type and speed, word-to-word changes in intonation and specific situational parameters like type of meeting (business versus personal), location (seedy versus safe), speaker and listener status (doctor versus patient) etc. Of course, the precision and utility of a computational representation often comes at the cost of a lower fidelity and greater abstraction. But in the case of untangling a complex phenomenon like the interaction between intent, context and behavior sacrificing a bit (or a great deal) of complexity is often the only way to get started. In fact, identifying where a computational representation falls short in terms of fidelity is another step toward identifying the essential aspects of the domain under study.

5. Next Steps

The initial construction of the middleware is complete. It is now possible for a scenario developer both to annotate utterances with high-level communicative intent tags and to define the relationships between the tagged intents and the behaviors that realize the intents in specific contexts. Further, we have developed a capability to supply the scenario author a "thumbnail" view to see how the behaviors are actually rendered in the synthetic environment without having to stand up the entire TLT Mission Environment system for each small tweak he might make (Figures 5.1 and 5.2).

The focus of this initial development has been on software function rather than theoretical exploration. Thus, the middleware currently exposes only a limited set of intents, contexts and behaviors. As we move into the second half of this effort, our focus will turn to extending that set and exploring further the extent to which believable communicative behaviors can be realized in the synthetic environment. We believe that the approach of dealing separately with communicative intent, context and behavior, will provide the kind of scalability and divide-and-conquer benefits that will make this exploration relatively painless.

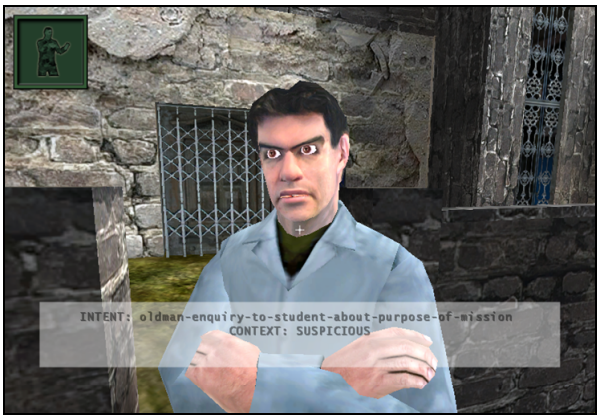

Figure 5.1 The communicative act of enquiring about your mission delivered by a character in the context of great suspicion.

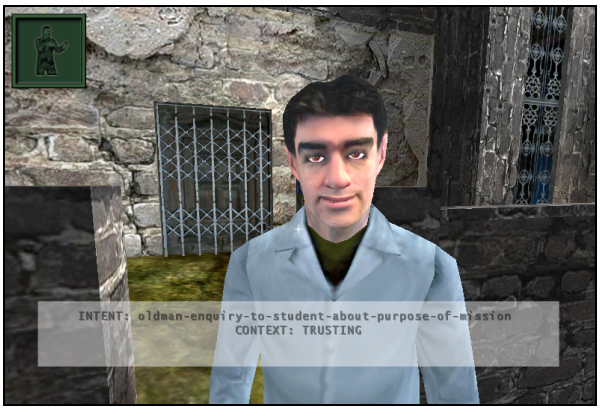

Figure 5.2 The same communicative act, but now delivered in a trusting context. The context and corresponding behaviors are selected at run-time once the mission is launched.

6. References

Cassell, J., (2000), "Nudge Nudge Wink Wink: Elements of Face-to-Face Conversation for Embodied Conversational Agents", in (Cassell et al. eds.) *Embodied Conversational Agents*. Cambridge, Mass: MIT Press.

Cassell, J., Vilhjálmsson, H., and Bickmore, (2001), "BEAT: the Behavior Expression Animation Toolkit", *Proceedings of ACM SIGGRAPH 2001*, Los Angeles, August 12-17, p.477-486.

Johnson, W. L., Marsella, S., Vilhjalmsson, H. (2004), "The DARWARS Tactical Language Training System", *Proceedings of the Interservice/Industry Training, Simulation and Education Conference*, Orlando, FL

Marsella S. and Gratch, J. (2003) "Modeling Coping Behavior in Virtual Humans: Don't Worry, Be Happy", *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, Australia

Marsella, S. and Pynadath, D.V. (2004) "PsychSim: Agent-based modeling of social interactions and influence", *Proceedings of the International Conference on Cognitive Modeling*, Pittsburgh, PA

Author Biographies

WALTER WARWICK is a Senior Research Analyst at Micro Analysis and Design. He is working on several projects having to do with the modeling and simulation of human behavior. He received his Ph.D. in History and Philosophy of Science, an Area Certificate in Pure and Applied Logic, and an M.S. in Computer Science from Indiana University.

HANNES VILHJAMSSON is a research scientist in the Center for Advanced Research in Technology for Education at the USC Information Sciences Institute. His research focuses on nonverbal cues in face-to-face interaction and how they can be supported in virtual environments. He received his Ph.D. in Media Arts and Sciences from MIT.

ACKNOWLEDGEMENTS This project is part of the DARWARS Training Superiority Program of the Defense Advanced Research Projects Agency. The authors wish to acknowledge the contributions of the members of the Tactical Language team and the BCBM team in particular. Many thanks to W. Lewis Johnson and Stacy Marsella for their input and support.