MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Hosting Activities: Experience with and Future Directions for a Robot Agent Host

Myroslava Dzikovska

TR2002-03 January 2002

Abstract

This paper discusses hosting activities. Hosting activities are a general class of collaborative activity in which an agent provides guidance in the form of information, entertainment, education or other services in the user's environment (which may be an artificial or the natural world) and may also request that the human user undertake actions to support the fulfillment of those services. This paper reports on experience in building a robot agent for hosting activities, both the architecture and applications being used. The paper then turns to a range of issues to be addressed in creating hosting agents, especially robotic ones. The issues include the tasks and capabilities needed for hosting agents, and social relations, especially human trust of agent hosts. Lastly the paper proposes a new evaluation metric for hosting agents.

Proceedings of Intelligent User Interfaces 2002

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2002
201 Broadway, Cambridge, Massachusetts 02139

Submitted October, 2001; revised and released January 2002.

Hosting Activities: Experience with and Future Directions for a Robot Agent Host

Candace L. Sidner
Mitsubishi Electric Research Labs
201 Broadway
Cambridge, MA 02139
617-621-7594
csidner@merl.com

Myroslava Dzikovska
Dept of Computer Science
University of Rochester
Rochester, NY 14627
716-275-8479
myros@cs.rochester.edu

ABSTRACT

This paper discusses hosting activities. Hosting activities are a general class of collaborative activity in which an agent provides guidance in the form of information, entertainment, education or other services in the user's environment (which may be an artificial or the natural world) and may also request that the human user undertake actions to support the fulfillment of those services. This paper reports on experience in building a robot agent for hosting activities, both the architecture and applications being used. The paper then turns to a range of issues to be addressed in creating hosting agents, especially robotic ones. The issues include the tasks and capabilities needed for hosting agents, and social relations, especially human trust of agent hosts. Lastly the paper proposes a new evaluation metric for hosting agents.

Keywords: artificial intelligence, discourse, robotics, intelligent user interfaces, collaboration, hosting agents, embodied agents, collaborative interface agents

INTRODUCTION

Recent research on embodied agents has made it possible to create human-like animated figures which can interact with human users with speech understanding and generation capabilities for helpful tasks such as customer service representatives, tutors or real estate agents [6, 7,12]. In recent work, Cassell et al provided animated embodied agents with richer discourse and collaborative capabilities by combining Collagen [21] with the REA architecture for embodied agents [8]. This paper reports on using extensions of that architecture for a 3D agent, that is, a non-mobile physical robot, which operates as the agent in a tutoring application known as Paco [23]. The paper also outlines additional capabilities needed for the general class of hosting activities.

Hosting activities are a class of collaborative activity in which an agent provides guidance in the form of information, entertainment, education or other services in the user's environment (which may be an artificial or the natural world) and may also request that the human user undertake actions to support the fulfillment of those services. Hosting activities are situated or embedded activities, because they depend on the surrounding environment as well as the participants involved. They are social activities because, when undertaken by humans, they depend upon the social roles of humans to determine next actions, timing of actions, and negotiation among the choice of actions. Agents, 2D animated or physical robots, who serve as guides, are the *hosts* of the environment. This work hypothesizes that by creating computer agents that can function more like human hosts, the human participants will focus on the hosting activity and be less distracted by the agent interface. Tutoring applications require hosting activities; we discuss others later in this paper. We also describe our experience in creating a robot host for a tutoring application. Based on it, we explore a set of tasks and requirements for hosting agents and a issues in evaluation of hosting agents.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IUI'02, January 13-16, 2002, San Francisco, California, USA. Copyright 2002 ACM 1-58113-459-2/02/0001…$5.00.

EXPERIENCE with a Robot Host

Our experience in building a robot host relied significantly on the Paco agent [23] built using Collagen for tutoring a user on the operation of a gas turbine engine. Thus our agent took on the task of speaking all the output of the Paco system, a 2D application normally done with an on-screen agent, pointing to the portions of the display, as done by the Paco agent. The user's operation of the display through a combination of speech input and mouse clicks remains unchanged. The speech understanding is accomplished with IBM ViaVoice[tm]'s speech recognizer, the IBM JSAPI (see the ViaVoice SDK, at www4.ibm.com/software/speech/dev/sdk_java.html) to parse utterances, and the Collagen system to provide interpretation of the conversation, to manage the tutoring goals and to provide a student model for tutoring.

The Paco 2D screen for gas turbine engine tutoring is shown in figure 1. Note that the agent is represented by a small window, where text, a cursor hand and a smiling face

appear (the cursor hand, however, is pointing at a button at the bottom of the screen in the figure). The face changes to indicate six states: the agent is speaking, is listening to the user, is waiting for the user to reply, is thinking, is acting on the interface, and has failed due to a system crash.

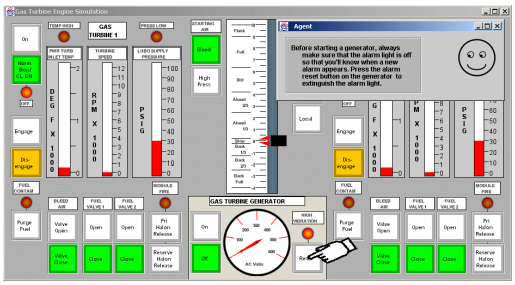

Figure 1:The Paco agent for gas turbine engine tutoring

Our robotic agent is a homegrown non-mobile robot created at Mitsubishi Electric Research Labs [Paul Dietz, personal communication], consisting of 5 servomotors to control the movement of the robot's head, mouth and two appendages. The robot takes the appearance of a penguin (whom we call Mel). Mel can open and close his beak, move his head in up-down, and left-right combinations, and flap his "wings" up and down. He also has a laser light on his beak, and a speaker provides audio output for him. See Figure 2 for Mel pointing to a button on the gas turbine control panel.

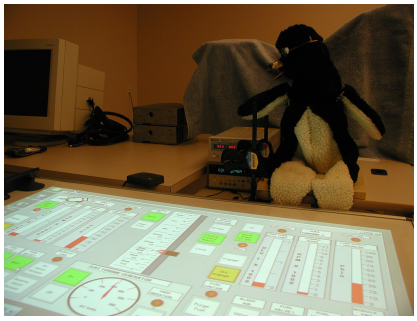

Figure 2: Mel pointing to the gas turbine control panel

While Mel's motor operations are extremely limited, they offer enough movement to undertake beat gestures, which indicate new and old information in utterances [9,19], and a means to point deictically at objects with its beak. For gas turbine tutoring, Mel sits in front of a large (2 foot x 3 foot) horizontal flat-screen display on which the gas turbine display panel is projected. All speech activities normally done by the on-screen agent, as well as pointing to screen objects, are instead performed by Mel. With his wings, Mel can convey beat gestures, which the on-screen agent does not. Mel does not however change his face as the onscreen agent does. Mel points with his beak and turns his head towards the user to conduct the conversation when he is not pointing.

Most of the challenges in using Mel were in creating all the necessary software so that Mel could take the place of the on-screen agent, get commands from the Collagen system and use the BEAT system for computing gestures synchronized with speech [9]. To conform to BEAT's requirements, speech generated by Collagen was marked up to provide information about clause boundaries, new and old information and deictic expressions in utterances. Deixis presented a special problem, because objects needed to be identified and translated to corresponding coordinates in the physical world. Since Collagen and BEAT used very different procedures for identifying the objects for deixis, we modified Collagen to provide the needed information.

Our experience points to the necessity of providing markup languages for multi-media input with a well thought out markup for deixis, not only to identify the objects properly, but also because pointing must be timed in some cases to coincide with the actual deictic linguistic expression. In Mel, word boundaries served as the synchrony points for speech and gesture, using events provided by JSAPI interface. For more realistic expression, with moving lips, phoneme timings need to be available.

Physical pointing also requires calibration of the servomotors of the robot to discover their coordinate systems with respect to the external coordinates known for objects in the world. Mel was carefully calibrated to provide the best possible pointing to the locations on the screen where buttons, sliders and other screen objects are represented. Mel's laser indicated just how accurate each pointing is, and while physical limitations of the servomotors made absolutely exact calibration and control impossible, we were able to achieve approximation good enough to provide the right gesture and location.

Mel would be a more effective host if it also had movable eyes, appendages more like arms and a vision system. Eye movements would allow it to convey the same basic information that on-screen Collagen agents do (which approximate the rather complex eye movements of humans), and arms would take on pointing so that the head could be used just for conversational turn taking and information structure. The need for vision is discussed later in this paper.

Mel is a direct product of extensive research on animated pedagogical agents [15]. It shares with those agents concerns about conversational signals and pointing as well as, in the future, the planning needed to locomote as part of its interactive behaviors. Unlike these efforts, Mel has greater dialogue capability, and its conversational signaling, including deixis, comes from combining the Collagen and Rea architectures. It also operates in the physical world. We contend that in addition to believability [1,15], hosts like Mel must have physical behaviors using vision and social behaviors. These are discussed later.

ARCHITECTURE of a Robot Host

The architecture of a Collagen agent and an application using Mel is shown in figure 3. Specifics of Collagen internal organization and the way it is generally connected to the applications are beyond the scope of this paper; see [21] for more information. The basic idea of the organization is that the application is connected to the Collagen system through the application adapter. The adapter translates between the semantic events Collagen understands and the events/function calls understood by the application. The agent controls the application by sending events to perform to the application, and the adapter sends performed events to Collagen when a user performs actions on the application. Collagen is notified of the propositions uttered by the agent via uttered events. They also go to the AgentHome window, which is a graphical component responsible in Collagen for showing the agent's words on screen as well as generating speech in a speech-enabled system.

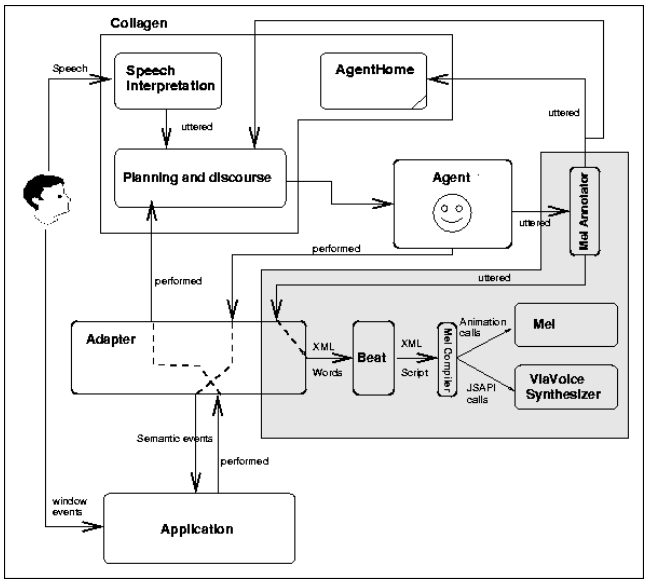

Figure 3: Architecture of Mel

In order to connect Mel to the Collagen system, we needed to make some changes in the system architecture. The shaded area highlights the components and events that were added to the original system. In the new system, the agent home window no longer generates spoken output, because Mel needs linguistic and location information in order to produce beat and deictic gestures, which is not available to Collagen. Instead, the generation is handled through the adapter, since it is the only part of the system connected to the application and thus aware of the actual locations of objects that need to be pointed at.

All text generated by the Collagen agent is passed through Mel Annotator. It adds the linguistic annotations necessary for BEAT to operate. Ideally, all this information would be added during the generation process. However, these annotations are done separately for two reasons. First, certain implementation details of the Collagen generation, which is currently not syntax-based, made it difficult to put the algorithms for marking up theme and rheme (i.e. linguistic markings of new and old information) inside the generator. Additionally, this design makes the system more modular, allowing Mel to be switched on and off as necessary, and enabling implementation of different annotators to reflect different possible heuristics in determining theme and rheme.

Consider the following comments from Mel to a student:

(1) Before starting a generator, always make sure that the alarm light is off, so that you'll know when a new alarm appears.

(2) Press the alarm reset button on the generator to extinguish the alarm light.

The Collagen generation for (2) produces the annotation shown in figure 4a). It includes the annotation (*NEW*) that "the alarm button" is a new entity in the conversation. After being processed in the annotator module, this utterance description changes to include theme and rheme material as shown on figure 4b)

While we could have used theme/rheme tagging inside of the BEAT toolkit, the richer semantic information available from Collagen (such as new entities) produced better markings of theme and rheme than the syntax-only ones provided by BEAT.

Once the linguistic information was added, the result is passed to the agent home window, which displays utterances in the Agent window, and to BEAT for generation through the application adapter. The adapter adds pointing information to the annotation and passes it to the BEAT pipeline. For utterance (2) above, the description is enriched to include the description in 3c) which tells BEAT that pointing is a high priority gesture and to point on the screen at the specified x/y coordinates.

The pipeline for the remaining portions of the system is the default BEAT pipeline provided with the toolkit [7], which

(a) Syntactic annotation from Collagen

<*clause*> <*verb*> Press </*verb*> **<*NEW*>** the alarm button **</*NEW*>** on the generator **</*clause*>** to **<*clause*>** <*verb*> extinguish </*verb*> the alarm light **</*clause*>.**

(b) Output of rheme tagging algorithm

<*clause*> <*verb*> Press </*verb*> **<*RHEME*>** **<*new*>** the alarm button **</*new*>** on the generator **</*RHEME*>** **</*clause*>** to **<*clause*>** **<*RHEME*>** <*verb*> extinguish </*verb*> the alarm light **</*RHEME*>** **</*clause*>.**

(c) Pointing information added

<***Point*** priority=5 locationx=556 locationy=436>

Figure 4: Mel annotation process

adds the head nods and beat gestures, and resolves conflicts between different gestures. For utterance (2), BEAT first adds a beat gesture (a kind of emphasis to indicate new material in the conversation) and a head nod. However, the head nod is filtered out because it conflicts with using the head to point at the screen location, and pointing is higher priority than head nods as beat gestures. The output of Beat pipeline is an animation script for (2), shown below. The script specifies speaking actions, pointing start and stop actions and gestures actions (with the right wing) to indicate a beat gesture at the new information of "the alarm reset button" with *WI* denoting the word index of the action

```
<Animationscript Speaker="Agent" Hearer= "User">
<Start Action="Speak"
  Speech="Press the alarm reset button on the generator to
  extinguish the alarm light." WI="0">
<Start Action = "Point" Priority = "5"
  Locationx = "556" Locationy = "436" WI="0">
<Start Action = "Gesture_Right" Priority = "1
  Typ ="Beat" WI="1">
<Stop Action = "Gesture_Right" Prioriy = "1
  Type="Beat" WI="5">
<Stop Action = "Point" Priority = "5"
  Locationx="556" Locationy="436" WI="13">
```

The Mel compiler, developed at MERL, converts the script into a sequence of Collagen and Mel calls to schedule a point action, and a beat action, and execute them by starting the Collagen vocalizer with an event-based scheduler attached:

```
Mel.SchedulePointAction(0,556,436)
Mel.ScheduleBeatAction(1)
Vocalizer.speak("Press the alarm reset button to extinguish
the alarm light, Mel)
```

JSAPI provides events as each word is spoken, and Mel will execute scheduled actions at corresponding word indices. Currently we only schedule "start" actions, because Mel control servos do not allow us to precisely control the duration of movements. If a better robot were available, the scheduler will also have to include the action end indices in the scheduling algorithm.

Since the design described above is fully modular, it allows us to easily add a hosting agent to any application run with Collagen, and to debug Mel and applications separately as needed.

Adding Mel to the Collagen architecture was fairly straightforward as it did not require major architectural changes. However, the simplicity of the current generation mechanism significantly limits Mel's pointing behavior. In the utterance "these are the 3 buttons and an RPM gauge," Mel should point first at the buttons (while saying "buttons") and then at the gauge (while saying "gauge"). However, there is no way to indicate in generation that the pointings are associated with particular references in the utterance, that is, that deictic acts are part of larger acts, some of which can be communications, and need to be associated in detail with them. Without this capability, the closest Collagen can come is to create a single step that has something to be said and some objects to point at. However, the generation will not be assured of linking the correct deictic gesture with each spoken phrase. This lack in linking becomes evident later in the pipeline because BEAT will not have enough information to script the deictic gestures with the individual phrases in the utterance.

The proper solution to this problem lies in sophisticated generation that treats reference as a communicative act and allows gestures with each act. Generation must also allow communicative acts to be combined, so that one utterance contains multiple (reference) communicative acts and their associated gestures. This type of generation would require a tighter integration of Collagen and Beat's generation to allow them to cooperate in deciding on pointing acts and synchronizing them with speech. Generation with these properties is well within the range of state of the art generation systems.

At present we have not evaluated Mel formally with users to determine whether its actions provide useable information and do not distract users in some way. However, our informal demonstrations indicate that users are able to follow Mel's pointing and are able to use the tutor screen interface readily.

HOSTING agents of the future

In the remainder of this paper, we consider a number of general issues that are necessary to extend our current robot, architecture and algorithms to creating a general framework for hosting agents.

TASKS for Hosting Agents

Previous hosting agent tasks conceived in the research community concern information presentation in on-line displays, such as maps for navigating a city, information on sample objects for use in sales and tutoring applications.

Given the already sophisticated development of some of the 2D hosting agents, why undertake the creation of 3D ones? After all physical agents, i.e. robots, offer more control challenges than their 2D counterparts. In the 3D world, there are already instances of hosting agents for entertainment and commercial products (such as the talking Barney doll for use with on-screen children's programs) [13]. These first generation hosting agents are dolls, but can produce speech output, have some kind of self movement (perhaps only to convey excitement rather than object interaction) and are aware of what the display conveys. They can also have limited speech input. Second generation robots, some for hosting and other related matters, [3, 4] have rudimentary human interaction capabilities. These robots, designed by the researchers in robotics, traverse space very successfully, but use limited

speech and some interface graphics via a laptop attached to the robot to interact with a user.

We are interested in activities characterized by:

- Sophisticated communication needs, including the use of vision, conversation and locomotion;
- Activities and objects with significant knowledge structure and the need for planning mechanisms in using the activities;
- The agent host needing to understand the user's purposes in order to collaborate (c.f. 11, 17, 22, 23), to perform actions and to request actions on the part of the user.

One such activity, which we are currently exploring, is hosting a user in a room with a collection of artifacts. In such an environment, the ability of the host to interact with the physical world becomes essential, and justifies the creation of physical agents. Other activities include hosting as part of their mission: sales activities of all sorts include hosting in order to make customers aware of types of products and features, locations, personnel, and the like; tutoring also includes hosting. In these activities, hosting may be intermingled with selling or instructional tasks. Activities such as tour guiding or serving as a museum docent are primarily hosting activities.

Hosting activities are collaborative because neither party determines completely the goals to be undertaken. While the user's interests in the room are paramount in determining shared goals, the host's (private) knowledge of the environment also constrains the goals that can be achieved. Typically the goals undertaken will need to be negotiated between user and host. Tutoring offers a counterpart to room exploration because the host has a rather detailed private tutoring agenda that includes the user attaining skills. Hence the host must not only negotiate based on the user's interest but also based on its own (private) educational goals. Accordingly the host's assessment of the interaction is rather different in these two example activities.

This work disregards agents that serve as a conversational partner in using an on-screen application, such as the agent for scheduling TV programs [10], where the user may or may not be able to manipulate the interface with a cursor. These agents are not hosting agents because they can be perceived as part of the actual on-screen application, even if their developers did not intend this perceptual integration. While the line between hosting agents and application agents may be fuzzy for on-screen activities, in the physical world, a robot is a distinct entity from the other items that it may be providing guidance about.

Physical presence makes it possible for a host to convey information and offer guidance in new and effective ways. Room hosting, and other real world hosting environments, require or encourage the user and host to move around, interact with objects, describe procedures, and for the host, assess the user's seeing and doing. Deictic gestures, i.e. pointing at objects with or without speech, help the user locate material quickly. Virtual agents, such as Cosmo [15], already combine deixis and locomotion in virtual worlds.

However, no virtual agents are currently able to use vision in the deictic process, even though it can improve pointing. It can be used both to detect the objects the user is currently pointing at, and to make the agent's pointing more precise by providing visual feedback to control the gesture. Currently, Mel requires time-consuming calibration every time it is moved. The use of vision to detect the objects in the room and re-calibrate the pointing automatically when the robot is moved or moves itself would improve its usability. Re-calibration is, in fact, within reach of current vision technologies. We have developed a simple application that can detect a laser pointer location and estimate the necessary adjustment to it based on camera picture of the screen. It currently requires a completely darkened room, which is impractical for our applications, but improving it and incorporating it into Mel is feasible in future work.

CAPABILITIES of Hosting Agents

The types of activities described above require a number of agent capabilities, both in terms of lower-level capabilities provided by the underlying software and hardware and higher-level behavioral strategies that build upon them. The host needs to have the following capabilities at its disposal:

- Producing and understanding conversational spoken language,
- Knowledge of the users' tasks, in some cases, knowledge of the rules of the game,
- Support for entertaining private as well as shared goals, and private motivations of the agent,
- Locating, pointing to and manipulating objects (with visual mechanisms),
- Recognizing and tracking the user, and recognizing location of user gaze,
- Reasoning about plans, movement in space, and replanning of plans due to goal changes evidenced from sensor data.

Based on these capabilities, the host needs to build the following strategies crucial for hosting tasks:

- Engaging users' interest and maintaining interest;
- Relating to and affecting the social status of the user;
- Projecting different social status and rules for the agent;
- Managing trust and authority issues arising from the interaction.

Among the capabilities above, several concern robots. Robotic agents must not only move about but also have visual sophistication, which the state of the art in vision is

now able to provide. Tracking user gaze is necessary not only for turn taking in conversation, but for understanding the current user interest. Pointing behaviors are more than just natural; they make communication efficient because pointing gestures can quickly convey the intended object without a long, complex description [15].

Agents must be able to maintain private as well as shared beliefs and goals in the interaction. Private knowledge allows the agent to assess its own contributions to the interaction as well as attain private goals, such as the educational ones for tutoring, maintaining user interest and managing social relations.

Engaging user interest might be considered to be part of turn taking since the pauses and uptakes in turn taking can reflect user interest. However, engagement is as a kind of personal connection that goes beyond turn. Engagement is the process by which two participants in an interaction establish, maintain and end their perceived connection to each other. Engagement includes making initial contact with another, negotiating whatever collaboration between participants will occur, checking via sensor input that the other participant continues to be engaged, and evaluating if and when to break the engaging connection.

In hosting activities, a user may demonstrate lack of interest in what the host offers even after the two are engaged in an interaction. If the host intends to remain engaged with the user, and intends that the user also remain engaged, then the host must plan out a course of action either to recover user interest or to change its own goals and possibly the collaborative goals of the host and user. Agents such as Steve [14] check user gaze in interactions. However, keeping a partner engaged and negotiating that engagement is a new undertaking in agent behavior.

Social status and social relations between user and host cannot be overlooked in hosting. This paper focuses on two aspects of status: user trust of hosts and variety in agent social roles.

Human users must have a means of trusting hosting agents. Bickmore and Cassell [2,5] argue that social conversation (so called "small talk") develops trust among human conversational partners (such as real estate agents). However in performing tasks, when a host requests action on the part of the human, and the setting is unfamiliar, dangerous or potentially embarrassing, trust plays an even more critical role. While Reeves and Nass [20] report that people already view even computer workstations as objects of social interaction, viewing a host as a social agent is not sufficient in difficult situations. Trusting the host is a necessary prerequisite to the performance of the tasks that users and hosts wish to collaborate on.

Little is yet known about the nature of users trusting computer agents. What does it mean to trust a computer agent? Is it an all or nothing commitment? Or do users offer trust in degrees? If trust comes in degrees, how are agents to access that partial trust users offer or display? How does an agent foster user trust?

When agents can converse with users, linguistic means to persuade, coax or coerce users into trusting an agent is possible. Is this effective? Katagiri et al [18] report that the persuasiveness of an (on screen) agent affects the user's willingness to interact and accept the agent's advice, and that the agent's ability to project an authoritative role affects outcome of user tasks. While Katagiri's tasks were not threatening ones, such tasks suggest authority as a fruitful role for agents. Other possibilities for developing trust include small talk or host demonstrations: hosts performing tasks themselves to convey the safety, ease, etc. of the task. All of these options are in fact required depending on the type of user, the criticality of the task, time available, physicality of the host, and history of the interaction.

Robotic hosts raise a whole new array of questions in terms of trust and authority. What will be required from a robotic host to be perceived as authoritative? Is authority culturally determined? Users tend to describe our simple robotic agent as "cute". How would that impact both their trust in it and their willingness to cooperate? One study [24] shows that users trust agents depicted with computer simulations of animal faces and animations of animal faces more than agents with human face animations; but users cooperated more with the human face agents. Would that be still true for a physically present agent? These questions are a topic of on-going research.

Hosts cannot always act authoritatively. Sometimes they need to switch to roles/relationships that are of lesser or equal status with the user. Hosting activities can include behaviors that can be characterized as entertaining, e.g. making jokes, offering witty comments or acting in clown-like ways. These behaviors can serve to keep the user's attention, draw attention back when it has been lost or lead to empathy from the user (which is relevant to persuasion). Hosts may also benefit from adapting their status relative to the user over time based on observations about user interest.

Neither Collagen nor Beat offer consider social roles or trust in collaborative activities. At present we see these being included in the hosting architecture by means of models that are updated as the conversation (and sensors from vision and audio) provide information that is relevant to social roles and measures of trust. The real challenge in including social roles and aspects of trust is to create models based on adequate theory of human interactions which must then be applied to hosting agents. Walker et al [25] provide a model to vary linguistic utterances based on the maintenance of the social face of interlocutors. This type of model could be adapted to update and vary social roles by adjusting social role rather than social face, on the basis of both linguistic styles of utterances and decisions by the agent to alter social roles due to perceived changes in engagement.

In sum, hosts need vary their perceived status in relation to users in order to engage users, manage user task performance and encourage information sharing. To perform in this way, they must have at least implicit models of social status, and these models must be detailed enough to permit a change in status to occur between individual actions that are part of the interaction.

REASONING for Hosting Agents

The capabilities of hosting agents described above point to the requirements of the reasoning system needed to support them. In particular, agents need to be able to reason about beliefs, desires, intentions to and intentions that (for discussion of these distinctions, see [11]). Agents will have private beliefs, in particular about plans for providing services for users, desires for how to engage users as well as intentions to do so. Because the agents will act collaboratively, they will need to have shared beliefs and collaborative plans.

The above capabilities can reasonably be required from any intelligent collaborative system. However, a hosting agent, especially a physical hosting agent, will require additional capabilities to be most effective. In particular, it should be able to reason about pointing, moving, talking about entities and objects, and gazing at and manipulating objects. While current animated agents such as Steve and Cosmo [15] plan out these actions, their "movement" is accomplished by animations. Physical hosts make the leap to the limitations of the physical world and will require even more subtle reasoning than exhibited in previous agents.

In our current implementation of Mel, the communicative non-verbal behaviors (i.e. gaze, head nods, beat gestures) are generated by Beat, which is not aware of the state of the task, nor user's and agent's intentions and plans. As our experience showed, this may not be entirely adequate in many intelligent applications. Consider the case for deixis. The default model implemented in Beat is to point at every new object co-present in agent's and user's physical space. This is a good general strategy. However, in a tutoring application it would not produce the pointing gestures produced by the PACO agent when, for example, a user is confused and needs to be directed to a button he already saw before. Therefore, the agent needs to generate the relevant pointing behavior itself and needs to be able to reason about how to do it appropriately. A limited form of such reasoning is implemented in Collagen by including pointing acts in tutoring recipes. However, a more general theory of deixis as well as special reasoning rules about pointing is needed to provide more natural deictic gestures.

EVALUATION of Hosting Agents

How do we determine if a hosting agent is performing in a useful or valuable way for a user? Is it only that the agent and the user get their collaborative task done? Is user satisfaction with the interaction the only other key criterion? While completion of task, time to completion, and user satisfaction are relevant, necessary measures of hosting agent value, a third measure, which we will dub *normative behavior*, offers a measure associated with the agent itself.

It is a hypothesis of this work that the more effort a user must expend on understanding the interface and how to use it, the more time, attention, and effort the encounter requires. When interacting with computer hosts, whether human-like or creature-like, the user can either make use of his or her understanding of human host interactions or take the time and effort to understand a new way of interacting with the computer host. Clearly our goal is to create a computer host that can interact more like a human host does in order to reduce the level of effort needed to interact. To measure the computer-human hosting situation, we propose to use the human to human hosting situation as a baseline normative behavior. The more computer host actions and communications deviate from the human ones, the more the user behavior will likely deviate as well. Thus, for example, if the user must ask more questions, lets his/her attention wander more, or spends more time with the agent than a user does with a human host, the user's behavior is an indication of the quality of hosting behavior. In sum, by measuring features of the human baseline, and comparing the human-computer interaction to them, a measure is obtained for the quality of hosting that goes beyond task success and time to completion.

It is not simple to gather normative behavioral data nor to judge deviations on the part of hosting agents. However, previous research on hosting agents illustrates a number of features of interactions that are well enough understood to serve in defining this measure. They include the use of visual cues of user attention, and judging aspects of conversational interaction, such as turn taking behaviors, focus shifts in relation to the task [16], pertinence of answers to questions, and frequency of negotiations of next task. These features provide an initial collection on which to compare normative human behavior to the computer host's.

CONCLUSIONS

Hosting activities and hosting agents provide a class of multi-modal interfaces that transcend environments organized around computer displays and GUI interfaces. They require not only conversation and collaboration on tasks, but also social skills from the host to develop user trust, and robotic skills to observe the user in conversation and activity and to permit the robot to move about in serving as a host. While such requirements have been in the minds of AI researchers for decades and are in part addressed by research on animated agents, it is now possible to pursue such hosts because of advances in vision, robotics and collaborative activity. However, one must not just bring together these advances in a computer architecture (a challenge in its own right). It is also necessary to discover how to use planning techniques and

social roles to round out the capabilities of hosting agents, as well as to devise means of evaluating the value of the hosting agent. This paper has reported on our initial experience in creating hosting agents and has described directions needed in each of these areas in order to create hosting agents that users will want and will find valuable to interact with.

ACKNOWLEDGEMENTS

We thank our colleagues Paul Dietz, Neal Lesh, Doug Macclure, Chuck Rich, and Chris Wren. Thanks also to the anonymous reviewers of this paper for IUI-2002.

REFERENCES

1. Bates, J. Loyall, A. and Reilly, W. Integrating Reactivity, Goals, and Emotion in a Broad Agent. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pp. 696-70, 1992.
2. Bickmore, T. and Cassell, J. "Relational Agents: A Model and Implementation of Building User Trust". *Proceedings of CHI-2001*, pp. 396-403, ACM Press, New York, 2001.
3. Bruce, A., Nourbakhsh, I., Simmons, R. The Role of Expressiveness and Attention in Human-Robot Interaction. *Symposium on Emotional and Intelligent II*, AAAI Fall Symposium Series 2001, AAAI Press, Menlo Park, CA, 2001.
4. Burgard, W., Cremes, A. B. et al, The Interactive Museum Tour Guide Robot, *Proceedings of AAAI-98*, 11-18, AAAI Press, Menlo Park, CA, 1998.
5. Cassell, J., and Bickmore, T. Negotiated Collusion: Modeling Social Language and its Relationship Effects in Intelligent Agents. *User Modeling and Adaptive Interfaces* (to appear).
6. Cassell, J., Bickmore, T., Vilhjálmsson, H. and Yan, H. More Than Just a Pretty Face: Affordances of Embodiment, *Proceedings of IUI-2000*, pp. 52-59, ACM Press, 2000.
7. Cassell, J. Sullivan, J. Prevost, S. and Churchill, E. Embodied Conversational Agents, MIT Press, Cambridge, MA, 2000.
8. Cassell, J., Nakano, Y. I., Bickmore , T. W., Sidner, C. L. and Rich. C. Non-Verbal Cues for Discourse Structure. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, July 2001.
9. Cassell, J., Vilhjálmsson, H., Bickmore, T. "BEAT: the Behavior Expression Animation Toolkit *" Proceedings of SIGGRAPH 2001*,pp. 477-486, ACM Press, New York, 2001.
10. Cavazza, M. Representation and Reasoning in a Multimodal Conversational Character, *Second IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, IJCAI-2001*, Seattle, pp. 15-20, August, 2001.
11. Grosz, B. J. and S. Kraus, Collaborative Plans for Complex Group Action. *Artificial Intelligence*, 86(2): 269-357, 1996.
12. Hayes Roth, B. The Extempo Ambassador, available at http://www.extempo.com/, 2001.
13. Johnson, M.P., Wilson, A. , Blumberg, B. , Kline, C. and Bobick, A. Sympathetic Interfaces: Using a Plush Toy to Direct Synthetic Characters, *Proceedings of CHI-99*, pages 152-158, ACM Press, New York, 1999.
14. Johnson, W.L. and Rickel, J.W. Steve: An Animated pedagogical agent for procedural training in virtual environments. *SIGART Bulletin* 8:16-21, 1998.
15. Johnson, W.L., Rickel, J.W. and Lester, J.C. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education*, 11, 47-78, 2000.
16. Lesh, N., Rich, C. and Sidner, C. L. Collaborating with Focused and Unfocused Users. *Proceedings of the 8th International Conference on User Modeling*, Springer Verlag, New York, pp. 64-73, 2001.
17. Lochbaum, K. E. A Collaborative Planning Model of Intentional Structure. *Computational Linguistics*, 24(4): 525-572, 1998.
18. Katagiri, Y., Takahashi, T. and Takeuchi, Y. Social Persuasion in Human-Agent Interaction, *Second IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, IJCAI-2001*, Seattle, pp. 64-69, August, 2001.
19. Prince, E. Toward a Taxonomy of Given-New Information. In *Radical Pragmatics*, P. Cole (ed.), Academic Press, Inc. New York, 1981.
20. Reeves, B. and Nass, C. *The Media Equation. How People Treat Computers, Television, and New Media Like Real People and Places*. Stanford, CA: CSLI Publications, 1996.
21. Rich, C., N. Lesh, C. L. Sidner . COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction, *AI Magazine, Special Issue on Intelligent User Interfaces*, to appear in 2001.
22. Rich, C. and C. L. Sidner. COLLAGEN: A Collaboration Manager for Software Interface Agents. *User Modeling and User-Adapted Interaction*, Vol. 8, No. 3/4, 1998, pp. 315-350.
23. Rickel, J., Lesh, N. , Rich, C. , Sidner, C. L. and Gertner., A. Using a Model of Collaborative Dialogue to Teach Procedural Tasks. *Working Notes of AI-ED Workshop on Tutorial Dialogue Systems*, San Antonio, TX, pp. 1—12, May 2001.
24. Parise, S., Kiesler, S., Sproull L. and Waters, K. My Partner is a Real Dog: Cooperation with Social Agents in *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work.* ACM Press, New York, pages 399-408, 1996.
25. Walker, M., Cahn, J. and Whittaker, S. Improvising Linguistic Style: Social and Affective Bases for Agent Personality. *Proceedings of the First Conference on Autonomous Agents*, pp. 96-105, ACM Press, NY, 1997.