Chapter 2

AGENT-AUGMENTED MEETINGS

Clarence Ellis
Computer Science Department
University of Colorado, Boulder, USA
skip@cs.colorado.edu

Jacques Wainer
Institute of Computing
State University of Campinas, Brazil
wainer@ic.unicamp.br

Paulo Barthelmess
Computer Science Department
University of Colorado, Boulder, USA
barthelm@cs.colorado.edu

Abstract This chapter presents the Neem Project, a research project that integrates intelligent agents and virtual participants into a distributed meeting environment. The agents incorporate knowledge about different aspects of a “good” meeting, or in the terms of the paper, they embody different models of how a meeting should proceed, and interfere in the meeting when they feel that it is appropriate to correct its course. Agents with similar goals are grouped into the same virtual participant, which channels the agents contribution through a single animated character. The implementation of different agents is discussed, as well as the software platform that allows for fast prototyping of different agents, and distributed meeting tools.

Keywords: agents, meetings, distributed meetings, virtual participants

1. Meeting support

Meetings are ubiquitous. They come in many sizes, shapes, and flavors. Organized business meetings are one form of interaction that can at times serve the purposes of informing participants, of mediating differences of opinion, of gathering expertise to make informed decisions, and of allowing participants to have better understanding of other participants and their context. If humankind is a fundamentally social animal, then meetings serve not only technical purposes, but fundamental social purposes. We know that social interactions during meetings are important; are tightly interwoven with business; and are critically important to success of meetings.

However, many meetings do not live up to the expectations and hopes of their participants. The literature has identified numerous problems and breakdowns that can occur within meetings. Many meetings are fraught with problems. The average business manager and technical professional spend nearly one fourth of their total work week in meetings. And the frequency of meetings are growing. Studies suggest that over 50% of the productivity of typical business meetings is wasted (Gordon, 1985). In a Bell Telephone Lab study, technical engineers frequently said that meetings were one of their biggest wastes of time. They wished that meetings would stop interfering with their work. By improvement of the business meeting venue, they might discover that productive meetings are an important component of efficient, effective work.

1.1 Related work

There is a lot of literature discussing problems and some potential solutions of meeting failures and fiascos (Hentschel, 1984; Robert, 1915). Poor meeting planning, poor organization and execution, and poor meeting follow-up are cited as causes of ineffective meetings.

Meetings have also been one of the main concerns of CSCW and groupware research. There has been a large set of research and prototypes developed within the CSCW community to support and augment distance meetings. These prototypes and products are what Ellis and Wainer, 1999 calls **communicators**, that is program and infrastructure that allow for many forms of communications among participants in a meeting. The communication can be textual, both synchronous and asynchronous (chats and e-mail), or may be sound and image based. Some research augments the basic multimedia communication with awareness information and tools (Nakanishi et al., 1996).

A different research focus is the elaboration, development and implementation within a system of meeting methodologies that try to improve the quality of the meeting, especially the quality of the decision. The research in GDSS (DeSanctis and Gallupe, 1987; Whitaker, 1994) falls into this category. Methodologies for decision meetings, such as Delphi method, nominal group method (Linstone and Turoff, 1975), and others have been implemented into GDSS. The idea is that the meeting should follow a predefined sequence of stages, where each stage has different goals and possibly different supporting tools. The global goal is to improve the quality of the decisions, by for example, separating the generation of ideas from the discussion of ideas, so that participants will not feel constrained in proposing something that otherwise would be immediately criticized. Other GDSS tools try to limit the forms of contribution to a meeting, for instance in an issue-based system (Conklin and Begeman, 1998), in such a way that the contributions are organized in a semantically well defined way.

This research focuses on a different direction, of providing a set of intelligent agents that make contributions sometimes through animated characters, or as voice output, sometimes as interface widgets in the direction of improving the meeting effectiveness and group's well being. This research (the Neem Project) has its precursors in Project NICK (Ellis et al., 1986), which provided tools and what we will call today, agent support, for face to face meetings. The description of some of the agents to support a text-based, distant interaction were presented in Wainer and Braga, 2001.

Neem employs an opportunistic approach where the system dynamically adapts based on reasoning over a context of (mostly human-to-human) interaction. In this sense Neem is more closely related to the work presented e.g. by: Jebara et al., 2000 in which the system acts as a mediator of the group meeting, offering feedback and relevant questions to stimulate further conversation; Isbister et al., 2000, whose prototype mimics a party host, trying to find a safe common topic for guests whose conversation has lagged; Nishimoto et al., 1998 whose agent enhances the creative aspects of the conversations by entering them as an equal participant with the human participants and keeping the conversation lively; CMU's Janus project [http://www.is.cs.cmu.edu/mie/janus.html] is somewhat related, in its aim to make human-to-human communication across language barriers easier through multilingual translation of multi-party conversations and access of databases to automatically provide additional information (such as train schedules or city maps to the user). While Neem shares the interest in human-to-human mediation, its goals are more ambitious than keeping a bi-party conversation going. Neem targets social and cultural aspects and is therefore concerned with a more detailed view of how groups work, and how collaborative systems can contribute.

Our philosophy includes a belief in active meeting augmentation, and a belief that well researched, well thought out technology can truly make a significant positive difference. Specifically we think that different functionality and goals can coexist within a meeting. For example, there should be a healthy tension between fast pace (efficiency) and careful quality of interaction (effectiveness); likewise, between convivial agreement and critical questioning of assumptions. Different perspectives can conveniently be represented as different roles played by different participants - some of them automated agents. For example a socially minded agent may want to take time to enjoy side discussion that informs all; as opposed to an organizationally minded agent that may urge the group to keep on its scheduled agenda, and to keep on time.

2. A Scenario

The following scenario illustrates some of the agents we have developed, some we are developing, and some that are still in the wish list. The meeting is of the paper selection committee of a conference, who is evaluating each one of the submitted papers, and deciding whether they are accepted, rejected, or postponed for a following meeting.

The program committee meeting involves committee members who connect to the system from their home or office at a designated time. Simple desktop conferencing tools include whiteboard, chat tool, audio (and potentially video) tools, shared artifact viewers, a whisper function, and conversation tools. All participants view and manipulate various conference documents and lists. For example, upon login, all participants see a virtual conference room and conference table with icons of all participants that are present and their roles shown graphically. An agenda tool lists the items to be discussed during the meeting and time estimate bars for each item. As the meeting progresses, actual time bars accumulate next to each item. Thus participants can see at a glance how much time has been allocated to each item (and how much time was supposed to be allocated) in real time during the meeting.

At a particular point in the meeting, while discussing what appears to be a controversial paper, agent A interrupts the discussion and informs that it has searched 350 bibliographic web sites and found several documents on the exact topic of the submitted paper. None of the program committee members had been aware of these documents. This starts a long discussion on whether documents substantiate that this is a very important topic, whether that the topic is not a completely new one, and whether the topic should be considered as within the scope of the conference. After a while agent B privately warns the meeting chair that too much time has been spent on this topic and that the meeting must move forward. The chair asks agent B to warn her again after 10 more minutes if they remain in this topic.

Well before the 10 minute deadline agent C privately informs a few of the participants that if they have not made up their minds yet about how to classify the paper, they should bring up their doubts because the group is ready to vote on the paper. A few minutes later, C informs the chair that the group is ready to vote. Agent C (as well as agent B) interact with participants through an animated character that has the appearance of a male, which we will call Kwaku.

After the vote agent D suggests they take a 15 minutes break (through another animated character, that has a female appearance, which we will call Kwabena). A few minutes later the chair informs the system they will take a break. Agent E at this moment suggests through Kwabena a "get acquainted game" to be played by the participants during the break. The committee members laugh at this and decline.

After the break, roll call is again taken, and the evaluation of papers restart. From time to time, agent F privately tries to encourage some of the participants to comment on the particular paper that is being discussed. At a particular moment, agent F (using Kwabena) says to everybody, "Maybe we should hear what John has to say about this paper."

After another hour of meeting, the chair terminates the meeting. The remaining papers will be discussed in a next meeting which is schedule at that time. After the meeting is over, the participants fill out an online form that evaluates the meeting, and receive (from agent G) the minutes of this meeting, and the action list detailing what each one has to do before the next meeting. Each participant also receives from agent H via e-mail an evaluation of his role in the meeting, how the others perceived him, and suggestions as to changes in his behavior.

The Kwaku and Kwabena animated characters presented above are an important feature of the system. It is a working hypothesis of this research that agents should be grouped into what we call *virtual participants*, that play *roles* within the meeting. There are numerous possible groupings and a spectrum of levels of personification of agent output. The layered structure of our platform allows us to explore this spectrum. Kwaku is an organizationally-minded virtual participant: it groups and controls all agents that are concerned with the evolution of the meeting towards its goals. Kwabena is a socially-minded virtual participant which is concerned with the well being of the participants in the meeting. A third virtual participant, Kwesi, is the informationally-minded virtual participant, which was the character used to output agent A's contribution to the meeting.

2.1 How are these things accomplished?

Agent A above (the one that collected the bibliographic references) is a task specific agent, one that has knowledge and a repertoire of actions that are suitable for a particular task. This particular one searches bibliographic sites for papers that have a) the same author name as the author of the paper being discussed, and b) papers whose title contains expressions or words that are closely related to the title of the paper being discussed. Such agent must know the title and author of each paper being discussed and also the correspondence between paper and agenda item. So, if the 7th agenda item is currently in discussion, and that corresponds to paper X, the agent will search for information regarding X. This agent is programmed to search specific bibliographical sites, applying some heuristic to measure similarity between papers.

Agent A has also a model on how and when it should interrupt the meeting. In some cases it pops up a small information window on each participant's screen, in other cases it will speak (through Kwesi) to all participants.

Agent B (the one that warned the chair that too much time was spent on a topic) is an **agenda agent** - an agent that keeps track of the flow of the meeting in terms of agenda items. It also has a model or theory about when and who to warn if too much time is being spent on an agenda item. It knows in which agenda item the meeting is at, either because the chair has an interface widget that allows him to inform the agent which item is being discussed, or the agent can recognize sentences like "Let's move on to the next topic" or "let's move on to the next paper" spoken by the chair.

Agent C (which suggested to some people they should make up their minds about a vote) and D (which suggested a 15 minute break) are different **attitude agents**. Attitude agents collect information about the participants' attitudes towards the meeting, for instance "I am tired", "I am ready to vote", "I think this discussion is not advancing" and so on. Each participant expresses his attitudes by clicking on appropriate attitude buttons. Each attitude agent has a theory (or model) of how to handle attitudes expressed by participants, and who to notify about these attitudes. In particular agent C deals with the "I am ready to vote" attitude, and agent D deals with the "I am tired" attitude.

Agent E (which suggested the 'get-acquainted' game) is a **social well being** agent - an agent that has some model of how the group as a whole is behaving and the role each individual is playing in the group at each time. The model may be simple or elaborate. As an example of a simple (and naive) model, the agent may be programmed to suggest some form or other of a get-acquainted game if, as far as the agent knows, participants have never met before. Even this naive theory implies that some component of the system stores the history of previous interactions.

A more elaborate theory of social well being may be based on Bales and Cohen, 1979 Symlog Theory . Using Symlog, an agent may determine that a group is polarized between two positions, and that this is an undesirable situation. The agent can then propose some actions to try to reduce the polarization of the group.

Agent F (which said 'lets hear from John') is a **floor** agent that determines how much time each user has spoken during a meeting so far, and tries to encourage in different ways each less vocal participant to contribute. The agent simply collects data from a voice server on how much time each participant has spoken, and when some portion of the meeting has elapsed, ranks participants based on the amount of time they have spoken so far. For each participant the agent computes an encouragement index which is a function of many parameters. The less the participant spoke in relation to the average speech time, the higher the encouragement index. Also the meeting coordinator may have feed the agent with information regarding which participant's contribution would be more desirable in relation to each agenda topic, and so on. The floor agent also receives information from the agenda agent regarding which agenda item is currently being discussed, and the encouragement index is increased for those whose contributions are particularly welcome in that item, and decreased for the others, and so on.

After the encouragement index reaches a first threshold the agent sends encouragement messages to participants. If the index reaches a second threshold, the agent generates a public voice output on the lines of "Let's hear what X has to say about this". The floor agent has other means of action, such as subverting a strict FIFO order of the floor control, if that is represented as a queue, by moving participants with higher encouragement indexes to the front of the queue.

Agent G is a **minute in minutes** agent - an agent that summarizes the meeting. It collects all spoken contributions of each participant and transforms them into text, summarizes the resulting text and produces the meeting minutes at the end of the meeting.

Agent H is a **group analysis** agent. It embodies a theory of group behavior, and in particular a theory of **good** group behavior. By analyzing the evaluation each participant makes of others, the agent is able to compute a "summary" of the attitudes each participant exhibited during the meeting and from that, which role the participant played during the meeting. By itself this information may be important to the participants themselves; they can receive feedback by the system on their behavior and attitudes during the meeting, as perceived by others. Clearly, in almost all situations it is necessary for the evaluations to remain anonymous, so that the evaluators can make their evaluations without constraints.

There are very few theories that allow for a description of group behavior from evaluations of each participants' attitudes, or behaviors. Again, Symlog (Bales and Cohen, 1979) is one such theory, and a version of a group analysis agent based on the Symlog theory will be discussed (Section 4.3).

The virtual participants can be broadly described as a grouping of like-minded agents. All agents that are concerned with the well being of the participants are grouped under Kwabena. That implies the following:

- If an agent decides that it should say its contribution (as opposed to placing a widget on a participant's screen), the voice/image generated will be Kwabena's.
- If an agent has parameters that control its behaviour at run-time, those parameters are modifiable by a particular agent Kwabena-input. Kwabena-input responds to a user commands like "Kwabena, interrupt less" by setting the appropriate parameters of the agents grouped under Kwabena, so these agents will produce less output actions, and thus interrupt less.

At a deeper level each virtual participant's output may be tied to a particular agent (Kwabena-output) that performs a grouping of component agents. Kwabena-output allows the definition of Kwabena's "personality" or "point of view", by further reasoning, and filtering out some of the component agents' contributions. Thus if Kwabena has a more long term "point of view" towards meetings, it will filter out some of the contributions of agents such as C and D above, which have shorter term goals, and do not filter out contributions from agent E.

Agent A was partially implemented; the component that searches bibliographic references was implemented but the "agent" is not autonomous. It neither starts the search by itself, nor it interrupts with the results. In the terms defined in the next section, agent A was implemented as a tool and not as an agent. Agents B, C and D were implemented closely as to what was described. Agent E was not implemented at all: we have not yet found a suitable theory of social well being in a meeting. Our use of Bales' Symlog is embodied in agent H (group analysis) because we collect information about the meeting *after* the meeting, and thus there is no information during the meeting to propose any social action.

Agent F (floor agent) was implemented as a tool, not as an agent. It does not produce a voice/animated output, but places a histogram of all participant's speak time in the coordinator's screen. In the current version it is the meeting coordinator's task to encourage the less vocal participants to contribute. Agent G (minute in minutes) is not yet fully implemented. The agent only produces a transcription of each participant's contributions, but without summarization. Agent H is implemented as described.

3. Architecture: system and agents

3.1 Meeting system conceptual model

Our view of meetings is that they are interactions between various participants who play various roles. Roles are important. As described by Biddle, 1979, roles allow well understood division of work among a group, and mediate expectations of who will do what. Roles are a convenient mechanism for associating privileges (authority) and responsibilities to participants. Roles always exist within interacting groups, whether they be formal and explicit, or informal and implicit. Roles such as "devil's advocate" and "social matchmaker" are quite important although they are frequently informal and implicitly assumed by appropriate people.

In this research, meeting augmentation is accomplished through three complementary functionalities: tools, artificial participants, and agents.

Agents are intelligent components that embody some *ad hoc* theory or model for interpreting data, for analyzing the current situation of a meeting, for comparing it to what a "good" meeting should be, and for acting towards improving the meeting. Agents are described in more details in Section 3.2.

Tools are at the lower end of the agent spectrum, and correspond to components that embody a trivial model of how to interpret data, and a trivial model of how to display this data. The tool may display a list of participants waiting for their turn to speak, or the amount of time spent on each topic. In both examples, these tools receive mouse events as input, and do not need a complex model to interpret the data. In the first tool, a participant can click on a button when he wants to be added to the talk queue. The tool adds the user to it, and displays the talk queue as a widget in each of the participant's workstations. This tool could embody more intelligence, or in our terms, a more complex model of how to act, and would be better thought of as an agent. For example, the tool may have a model of "who should speak first," as discussed in Section 2.1, and reorganize the queue according to this model.

A key theme of this research is its use of active, anthropomorphized virtual entities as full-fledged meeting participants, the virtual participants introduced above. Virtual participants have visible characteristics called personalities, and their actions are driven by what is perceived as a set of consistent goals. Different virtual participants typically have different goals, which may be disjoint or overlapping.

For example, Kwaku has goals of keeping the meeting on time, and keeping discussions focused on an specified agenda. He keeps track of time, of who speaks, and of the agenda. If a meeting falls behind schedule, then Kwaku discourages informal chit-chat about peripheric topics. Kwabena has a goal of building common ground among participants, and knows that informal discussion about peripheric topics is important for this. Kwaku's and Kwabena's goals are therefore conflicting. Agents have voices and anthropomorphic appearances, so the result in this situation may be two spoken statements to all participants. In a strong voice, Kwaku might say, "Hey gang, we are behind time! Lets move on to the next agenda item." In a gentle voice, Kwabena says, "You all know it's important for us to have a common understanding of our team assumptions and direction. Whenever necessary, please take time to have clarifying discussions."

Some meetings will make use of all tools, agents and virtual participants, while others may be augmented only by tools and agents. It is as yet unclear which tools/agents/virtual participants are appropriate for each type of group and type of meeting. We will discuss in Section 6 issues related to the evaluation of the uses of tools/agents/virtual participants.

In the rest of the paper we will concentrate on agent-based augmentation functionality. We will not discuss tools at all and only briefly describe the virtual participants.

3.2 Agent architecture

In general one can classify agents in this system as *passive* or *active*. We will describe in detail the active ones, and show that passive agents are an important particular case of the more general active ones.

In general terms, social active agents gather data regarding the climate of a group's interaction, either from the content of the interaction itself (e.g. from spoken or written exchanges) or from private information provided by group members. The agents interpret the data, and compare it with a normative theory, that is a theory that describes how "good meetings" should proceed. The agents may compute some corrective actions once deviations are detected between what is perceived as being the climate of a meeting and what is established as desirable by the normative theory that drives the agents. Once a corrective action is determined, the form in which the action should be presented is also determined. Figure 1 illustrate the active agent architecture.

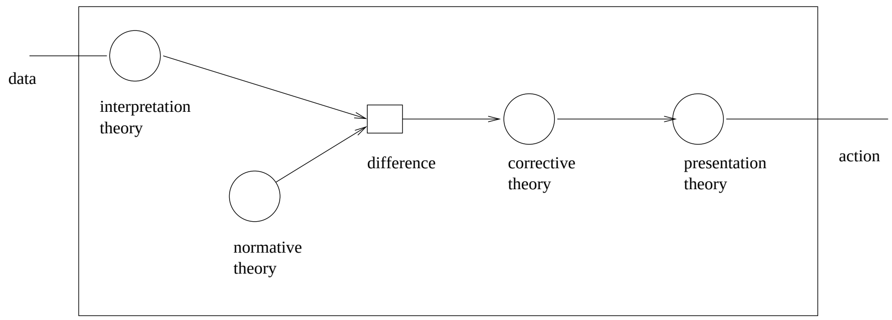

Figure 1. Generic active agent architecture

A passive agent does not embody a corrective theory and may not even embody a normative theory, so it can provide at most some form of feedback to the group members, in the hope that they will themselves make good use of such data. The feedback may consist of the interpreted data itself, or it may consist of the deviation between interpreted data and what the normative theory predicts. Again, passive agents may also have to reason on how to present the feedback data. Figure 2 illustrate alternative architectures for the passive agent.

Let us illustrate the architecture with a rather naive, made-up social theory. The theory states that in a brainstorming-like meeting the participants should speak an equal amount of time, i.e., a "good meeting" is one where all speak the same amount of time. Let us assume that the meeting has audio tele-conference facilities. By tapping into the audio server information, the agent can collect data about the total time each participant spoke so far. To interpret such data the system considers pauses of less than 2 seconds as "speech" and pauses longer than that as "silence". This is the interpretation theory.

One agent may display in an interface widget a chart of the "speech time" of each participant. This is a passive agent, with no normative theory (it does not have a theory of a "good" meeting), and no presentation theory (it does not reason about the most effective way of displaying

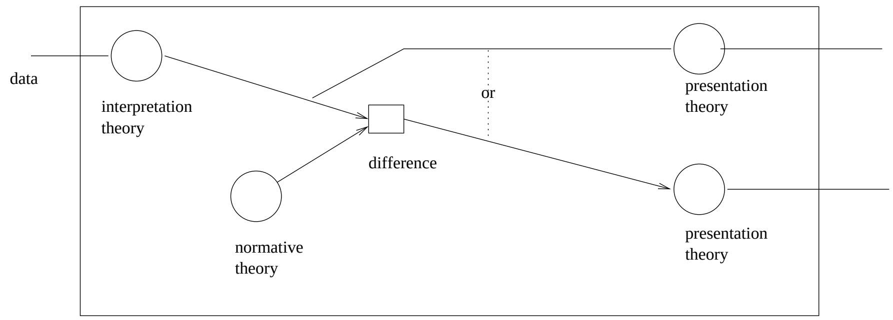

Figure 2. Generic passive agent architecture

the information). Probably just the common knowledge of such data will inhibit the more vocal members of the group and encourage the less vocal to speak more. One should notice that other forms of feedback are possible: an e-mail message to each participant after the meeting is over, containing just that participant's data, or everybody's data, and so on.

A different agent, which embodies a normative and corrective theory, would gage the "speech time" of each participant according to the normative theory that says that everybody should speak the same amount of time. This agent would then act on perceived differences between the normative behavior and actual behavior of each participant. A more elaborate normative theory could take into consideration other aspects, such as total meeting elapsed time, how much variation in "speech time" is acceptable (say one can speak up to 50% more than his equal share of time), and so on. In any case, divergences between what is established by a normative theory and a meeting's actual data is passed on to the corrective theory.

Again many different corrective theories are possible. To illustrate, let us assume that the evaluation of the meeting data with respect to the normative theory determined that Adam (one of the participants) spoke more than his share of time, and Beth, and Carlos both spoke less than expected. A corrective theory may decide that the correct action is to alert Adam that he spoke too much; a possible complementary action is to inform Beth and Carlos that they spoke less than their expected time.

At this point, the presentation theory is called upon. Adam may be informed that he spoke too much by a large widget placed in his screen, whereas Beth and Carlos' alert might be conveyed by a much less vivid widget. A different presentation scheme would wait until a silence is detected, and generate a voice output (recognizable as being from the system and not from another participant) that says "Let's hear what Beth and Carlos have to say about this". Finally, if the turn taking mechanism is implemented via a request queue, another corrective theory may just adjust the priorities of Adam, Beth and Carlos, so that they have respectively lower, medium, and higher priorities to get the floor when they ask for it.

It is important to point out that in practice the corrective and the presentation theories work together and not sequentially, as the model suggests (for pedagogical reasons). The model shows them as sequential only to stress the fact that both the content and the form of the action must be reasoned about.

Agents may work in tandem, i.e., the output or intermediate computation of an agent might serve as input for another agent. In the example above, the result of computing the "speech time," that is, the intermediate result computed by the agent based on its interpretation theory, may be made available to other agents. Similarly the agenda agent in the scenario above might provide important information to other agents.

4. Some agents

We will discuss in detail the implementation and theories behind some of the implemented agents.

4.1 Attitude agents

The attitude agents allow participants to declare their feeling and sensations towards the discussion. A participant, at any moment during the discussion can state, by pressing buttons in the interface, that he thinks the meeting discussion is going too slow, or that it is too unfocused, or the opposite, that it is going too fast, and that it is too narrowly focused.

We developed *ad hoc* theories to deal with this attitude information. The theories take into consideration the percentage of participants declaring a particular attitude, how much time has elapsed since a participant's last declaration of attitude, and who should be informed of this general feeling towards the discussion.

In general, a participant can express his opinion regarding the **content**, the **focus** and the **speed** of the discussions. In each dimension the user can choose either extreme characterization. In relation to the speed, the discussion can be classified as **Fast** or **Slow**, in relation to the content it can be classified as **Boring** or **Interesting** and in relation to the focus, it can be classified as **Narrow** or **Wide**.

The speed refers to how well topics are being explored. A fast discussion is probably discarding topics without fully exploring them; a slow discussion spends too much time in each topic. The focus refers to the breadth of discussion. A narrow discussion is probably not covering all necessary topics; a wide discussion is bringing in too many topics. The content dimension is less well defined, and allows users to express more of a 'gut feeling' than an objective analysis of the discussion.

The ad hoc social theory that deal with these attitudes define the following factors:

- **validity interval**. For how long is such a manifestation valid? Since these mood manifestations reflect a momentary attitude towards the discussion, the manifestations must be valid for a limited interval. Clearly if the discussion is dynamic, and all users participate in the discussion frequently, such duration must be short. In this implementation, we parameterized the validity interval, and this parameter once set remains fixed throughout a discussion.
- **percentage**. How much is a "large number" of participants? How many participants must have expressed their mood towards some attribute before the social agent should act. Again, in this implementation, this percentage is parameterized and remains fixed throughout a discussion.
- **feedback**. Who are the participants that should be informed when the threshold is reached and how are they to be informed? After a "large number" of participants expressed their mood towards some attribute, who should be informed and how? In some sense, this determination defines a set of participants that are to be blamed for a situation and/or a set of participants that should act to correct an undesirable situation. Participants that are responsible for the current state of affairs should possibly be informed of that state as a form of feedback. Similarly, participants that are in a position to correct a situation very likely should be informed of the state as well.

In particular, regarding the feedback, we developed the following justifications:

- **fast discussion**. If s is the validity interval for the speed dimension, then all participants that contributed to the discussion during the interval s are informed that a large number of participants think the discussion is going too fast. The intuition is that participants that were directly involved in the discussion should receive feedback on this issue.
- **slow discussion**. All participants that did **not** contribute to the discussion during the interval s are informed that a large number

of people think the discussion is going too slow. The moderator of the discussion is also informed. In this case we believe that the moderator, plus the people that did not participate during the validity interval can correct this situation, by making their contributions towards speeding up the discussion.

- **narrow** or **wide** discussion. If f is the validity interval for the focus dimension, then all participants that contributed for the discussion during the interval f are informed, as is the moderator.
- **interesting** discussion. If c is the validity interval for the content dimension, then all contributors during the interval c are informed that enough participants think the discussion is interesting. We believe that this will serve as a positive feedback.
- **boring** discussion. The moderator is informed. We believe that since this manifestation can result from many hard to determine causes, only the moderator should receive this information.

It is important to point out that the manifestations are anonymous, in the sense that no one, not even the moderator knows who manifested which attitude.

4.2 Ready-to-vote agent

The ready-to-vote agent is a simplification of the attitude agent. It is our experience that in decision-making meetings where there is a final vote to decide on an issue, the knowledge of when a reasonable number of participants has already made up their minds is useful and may lead to a more satisfying meeting. It is common that supporters of different positions will continue to argue their points of view even when most of the participants have already made up their minds.

The ready-to-vote agent is simpler because there is no temporal issue, or in other words, the validity interval is infinite. Once someone declares that she is ready to vote, that declaration remains valid for the rest of the discussion (or until it is explicitly revoked by the participant). The issue of who is responsible and who should apply corrective actions is also simplified: being ready to vote is not a "bad" meeting state, thus there is nothing to correct and no one to blame. But there are other stakeholders in the situation, in particular the participants that are not ready to vote. These participants must be informed so that they can either declare they are ready or submit the reasons they are not ready to the group.

In particular we defined parameters so that, when 70% of the participants voluntarily declare that they had made up their minds, the system informs that fact to the other 30% and query them regarding their readiness to vote. The system encourages those that are not ready to vote to post their questions to the discussion. Finally the moderator receives a more detailed report on the number of people that voluntarily declared themselves ready to vote, the number of them that did so after being queried, the number that explicitly declared themselves not ready, and the number of participants that did not answer the query. Using this information, the moderator can set the deadline for the discussion, and the starting time for the voting process.

4.3 Symlog agent

The Symlog Agent is an implementation of the Symlog theory. The goal of this agent is to provide feedback information to the participants. In the case of the bargraph (to be introduced momentarily), corrective suggestions are also provided.

Symlog, or System for the Multiple Level Observation of Groups, (Bales and Cohen, 1979; Polley et al., 1988; Hare and Hare, 1996; Bales, 1999) can be understood as a theory about group dynamics and group personality. The system uses a set of methods for measuring the behavior of group members at different levels: perceptions, attitudes, values, concepts, non-verbal behavior, public behavior, and also content of the individuals' verbal communication (Polley et al., 1988).

In the Symlog System, all of the behavior and the content of a group's interaction is represented in a three-dimensional space. The three dimensions in terms of behavior adjectives, are:

- U/D - *upward/downward* (Dominant vs. Submissive) An individual that is classified as **U** is active and dominant in his actions. An individual that is classified as **D** is relatively quiet and submissive to the dominant members.

- P/N - *positive/negative* (Friendly vs. Unfriendly) An individual that is classified as **P** agrees with others and smiles while listening. An individual that is classified as **N** is critical, does not smile, and is not used to listen to others.

- F/B - *forward/backward* (Instrumentally Controlled vs. Emotionally Expressive) An individual that is classified as **F** is controlled and has his attention focused on the main task of the group. An individual that is classified as **B** expresses emotion and is not directly interested in the main task of the group.

The Symlog space uses these three dimensions to form a cube. This cube, the Symlog Cube, is divided into 27 cells and each cell, except the central one, is named as a combination of one, two or three dimensions. The 26 directions constitute the complete group of values and/or attitudes and/or behaviors defined by Symlog, which are used to classify all the actions of individuals in a group interaction. For example, in one form of evaluation - the rating method - each of the 26 dimensions are mapped into 26 different questions about the behavior of each participant.

The rating method is based on a form with 26 descriptions of behaviors, and each participant must in retrospect, evaluate how often each of the other participants exhibited each of the 26 behaviors. Each participant also rates himself twice: as he would like to be rated by others and as he thinks he was rated by others.

The **Field Diagram** (figure 3) and the **Symlog Bargraph** (figure 4) are the main feedback forms presented by the Symlog theory. The first pictures all group members according to the three behavioral dimensions. The Symlog Bargraph is an individual's evaluation by the other group members.

The field diagram not only summarizes each persons global evaluation but places them in relation to one another. From these relations it is possible to infer whether the group was polarized into subgroups or not, and what roles some of the participants played.

In recent developments of Symlog, Bales, 1999 concluded that it is possible to empirically determine an optimal profile for a meeting participant. Bales, 1999 discusses in details both the limits and the meaning of such "optimal profile". This optimal profile, represented by the `E` in the bargraph can be contrasted with the participants' profiles (represented by `X`'s) , and large discrepancies or differences can be clearly shown.

At the end of each discussion, each participant ranks other participants according to the Symlog rating method, that is, each participant evaluates how often a participant engaged in each of the 26 behavior/attitudes defined by the Symlog 26 directions. Each participant also rates himself twice: as he would like to be rated, and as he thinks he will be rated by others.

Given all the evaluations, the system generates both the average field diagram and the bargraph for each participant. From the field diagram the agent determines the role each participant played. Participants are informed of the role they played, and of the meaning of the role. If a participant is knowledgeable in Symlog she can assess the field diagram herself.

The system also presents the bargraph to each user and for the three directions for which discrepancy between the participant's evaluation

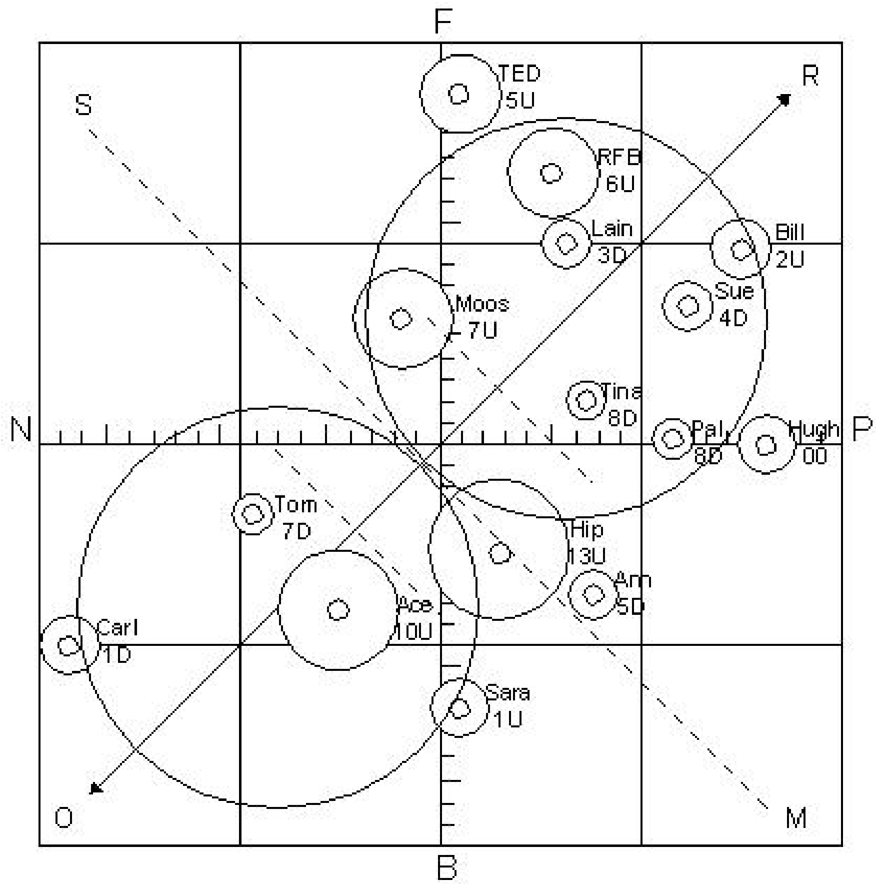

Figure 3. Symlog Field Diagram

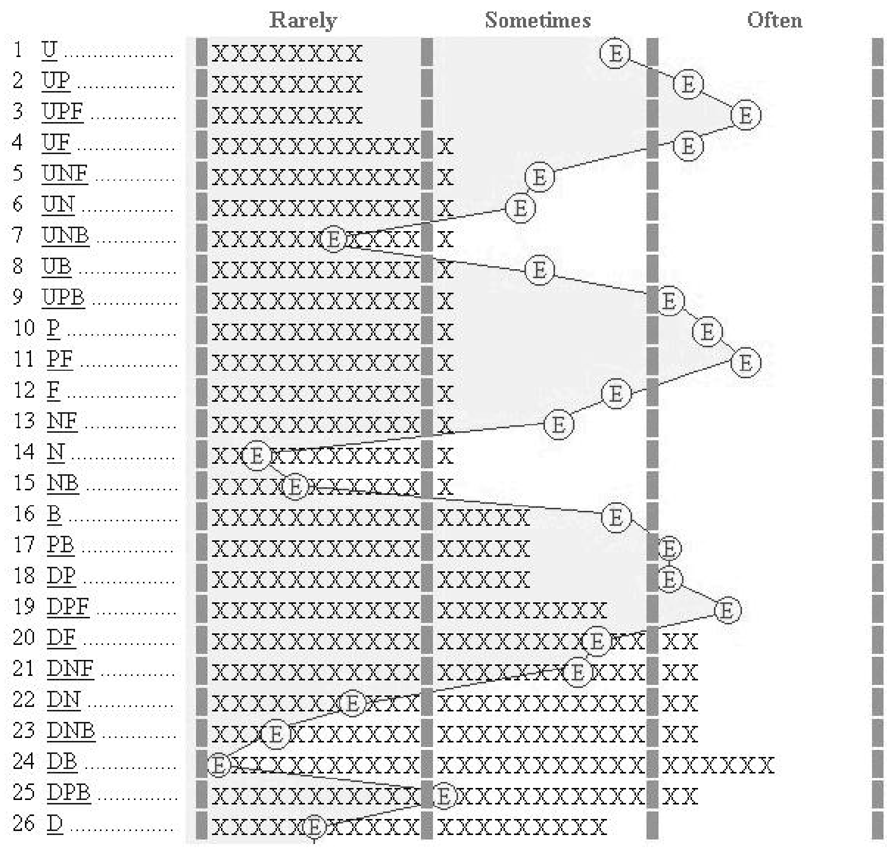

Figure 4. Symlog Bar Diagram

and the optimal level is the greatest, the system generates a text explanation and suggests that the participant should increase or decrease, according to the case, the frequency in which he engages in that behavior.

5. Implementation

Agents and tools are implemented as one or more components built on top of the Neem Platform. The Neem Platform is a generic framework for the development of collaborative applications. It is extensible, supporting rapid development of augmented synchronous distributed group applications. It embeds support for Wizard of Oz experiments - a traditional technique for testing new features in the field by having a human participant masquerade as the system, thus allowing a faster turnout than possible if everything needed to be coded. Further detail on the Neem Platform can be found in Barthelmess and Ellis, 2002.

The Neem platform is an infrastructure that binds together components that obey a uniform abstract representation. A message brokering infrastructure handles communication among components. Neem components are black-boxes that generate events and/or service events. Events in the system are reified as messages. Neem components can therefore be seen as message-enabled objects. A component signature consists of the messages it generates and messages that it services.

Components can be distinguished as *interface* or *augmentation* components. Even though conceptually similar (both are message enabled component types), *Neem Interface Components* (NICs) are characterized by their attachment to one or more interface devices, which makes them suitable for collecting and relaying interface events generated by each participant, in the form of standard messages. A *Neem Augmentation Component* (NAC), on the other hand, does not have this constraint and is purely a message processing device.

NICs provide means for the integration of multimedia devices, such as conventional monitor, keyboard and mouse consoles, audio and video. Other less conventional devices can also be integrated into a system through NICs, e.g. Virtual Reality (VR) goggles, haptik devices, etc. All that is required to integrate a new device is a set of NICs that interface with a device, extract events commanded by users and modify its state (for devices with output capabilities) according to commands received as messages. A NIC may for instance attach to an audio source (e.g. microphone) and do speech-to-text conversion, or extraction of prosodic features, or attach to a video source and do gesture or facial expression extraction. NICs also react to messages they receive, causing changes to the associated state of the interface, for instance rendering at users stations textual messages, graphics or full animations including gesture and/or voice.

Wizard of Oz functionality is supported straightforwardly by NICs that offer an interface through which a human participant can activates the generation of messages that cause other components to react. One can, for instance, send messages to components that control animated characters, making them move, speak, emote, and so on. Similarly, any other component can be made to react by issuing appropriate messages from a wizard interface.

Neem Augmentation Components (NACs) provide mostly back-end functionality, i.e., they are mostly responsible for processing the multiple modality streams, e.g. parsing natural language streams, *fusion, fission* of different streams and so on. NACs are also responsible for providing support for reasoning about the perceived context of an ongoing interaction and generating appropriate responses, according to the theories that inform agents.

The agents described in Section 4 are implemented by having their conceptual functionality mapped to one or more NICs and NACs. NICs allow for the collection of user events to which agents react, as well as the outlet of actions generated by these same agents. The implementation of the theories (interpretation, normative and corrective) can either be embedded in a NIC or most commonly, be implemented by one or more NACs.

Take for example the ready-to-vote agent (Section 4.2). Recall that this agent is responsible for detecting when 70% of the participants are ready to vote, and then inform this fact to the remaining 30% that have not made up their minds yet, to try to obtain faster convergence. This agent makes use of four NICs and one NAC. The first NIC corresponds to a tool that has a "ready to vote" button that is voluntarily pressed by those participants that feel they have made up their minds already. This button is a toggle, meaning that participants can at any time revise their readiness to vote by clicking on the button again to convey that something in the discussion made them feel it is not time yet to vote. A NAC monitors these button clicks and keeps a tally of the number of participants that have declared their readiness to vote at each moment. The NAC silently collects this information, up to the point when the parameterized threshold of 70% is reached; it then launches dynamically a second NIC, popping up a window only on the monitors of those participants that have not made up their minds yet. This second NIC informs participants of the fact that 70% of the participants are ready to vote and provides buttons through which queried participants can either declare that they feel ready to vote or not. For those that declare they are not ready yet, the NAC outputs a command that produces a voice message through Kwaku, encouraging them to post their doubts to the discussion. Kwaku's animation and voice generation is itself implemented as a NIC that can be commanded to produce arbitrary movements and voice output. Finally, a fourth NIC that can only be accessed by the moderator displays the data on voting readiness, informing the moderator how many participants voluntarily declared themselves ready to vote, how many did that after being queried, the number of participants that declared explicitly that they are not ready to vote yet, and the number of participants that did not answer the query.

For implementation reasons, the messaging infrastructure is implemented as two distinct environments - a *collaboration* and a *multi-agent* environment that are connected through a *coupler* component. The *distributed collaboration environment* provides support for participants' interaction and the *multi-agent environment* supports back-end augmentation functionality, such as multimodal processing and reasoning. Actual development leverages as much as possible on existing, field tested technology, based on open standards (Figure 5).

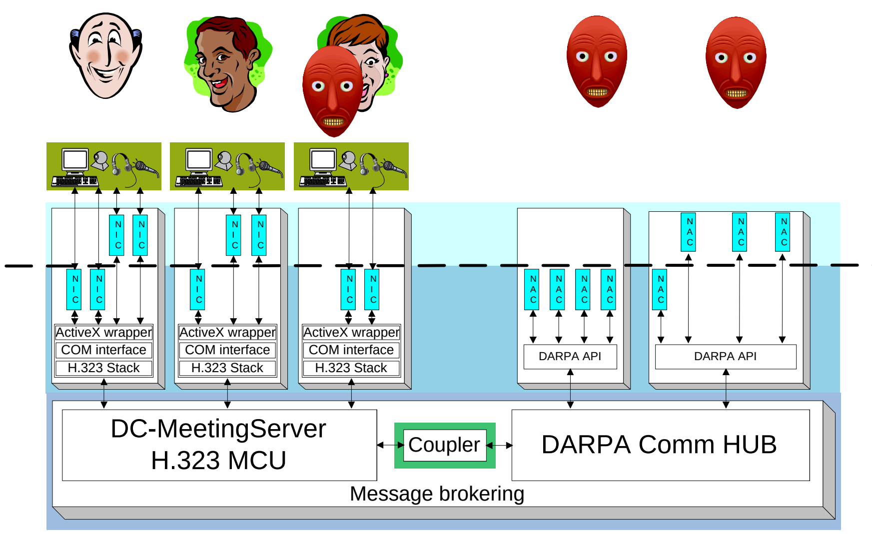

Figure 5. The Neem Platform.

The *distributed collaboration environment* provides message delivery to groups of distributed participants (or rather to the NICs through which they interact), either through broadcasts or selective delivery. It is also responsible for message serialization, to guarantee that participants' interfaces remain consistent throughout the interaction. Collaboration services are provided by DC-MeetingServer, a commercial H.323 Multipoint Conference Unit (MCU), produced by Data Connection Limited [http://www.dataconnection.com]. H.323 is a family of multimedia conferencing protocols published by the International Telecommunication Union [http://www.itu.org]. These protocols establish a set of services that can be employed as a basic multimedia conferencing support layer. It includes, among others, services for conference creation, handling client connection and disconnection, file transfer, whiteboard, application sharing, video and audio communication, and transfer of data between two or more connected clients. A variety of server and client software based on this protocol is readily available in many platforms.

The *multi-agent environment* is based on a blackboard-like architecture. The environment is organized in a hub-and-spoke configuration. The *mediator* (hub) controls the flow of information among other components (spokes). The mediator keeps a state that can dynamically influence the flow of information among the spokes. The spokes can trade information among themselves through the Hub. Spokes and hub can either be on the same machine or distributed. This is the environment that supports NACs. The multi-agent environment is built on top of the DARPA Communicator (which in turn is based on MIT's Galaxy architecture) [http://fofoca.mitre.org]. The Communicator is an open source hub-and-spoke architecture that provides a distributed, scriptable message passing system with special emphasis on building language-enabled dialogue systems. A Hub mediates connections between Communicator servers (such as speech recognition and synthesis, parsing, dialogue management, etc.).

Coupler is the component that binds these two distinct environments together - it translates between message formats and is responsible for: 1) relaying collaboration events to the multi-agent environment for analysis and 2) executing commands originated in the multi-agent environment, thus allowing for the sense that there are virtual participants taking part in a group collaboration.

The operational environment involves a variety of operating systems: Linux (running DARPA Communicator), Windows 2000 (running DC-MeetingServer), Windows XP (on the workstations).

Multimodal support in this initial phase consists of console i/o (monitor, keyboard, mouse) as well as natural language through typed and spoken messages. Natural language text output and animation, including voice production can be employed as output modalities, besides the activation of conventional widgets. Natural language processing capabilities running on the back-end are provided by language processing modules produced by Colorado University's Center for Spoken Language Research (CSLR) under the CU Communicator Project [http://communicator.colorado.edu/].

Currently, conventional interface components are developed in Visual Basic, C++ and Java. Speech-to-text is built on top of SAPI (Speech API). A variety of speech-to-text engines are compliant with SAPI. We currently employ IBM's ViaVoice 9.0's engine [http://www-4.ibm.com/software/speech/]. Animation is currently built using Haptek's VirtualFriends [http://www.haptek.com].

The Neem platform was used to construct agents described in section 4, a set of tools, a voice recognition component, and voice generation and animation generation for the virtual participants.

6. Future work and conclusions

The central hypothesis of this research is that meetings can be improved by augmenting them with tools, agents, and virtual participants. This paper has described the Neem Platform and some of the agents built using that platform.

This research is now moving into an evaluation phase: do meetings get better by the use of tools, agents, and virtual participants? With the help of social psychologists, we are currently setting up a series of laboratory experiments that try to get some answers to this question.

It is clear to us that different groups and different meetings need different forms of augmentation. A group that has social bonds among themselves, and have met many times before may only need unobtrusive, task specific tools. A group that is meeting for the first time may need socially-minded agents and Kwabena-like virtual participants. A large group with conflicting views may need organizationally-minded agents and Kwaku-like virtual participants.

Another dimension of exploration is the fine tuning of agents. Agents necessarily embody some arbitrariness in their models. Even if one agrees that a "ready-to-vote" agent is useful in large decision-making meetings in which participants may have conflicting views, it is likely that the specifics of the "ready-to-vote" model described in section 4.2 are debatable and need to be tuned for specific situations. Even if ready-to-vote agents are in general useful, it may be the case that with wrong settings they will cause more harm than good. That is even clearer for more pro-active agents, such as the floor agent suggested in section 2.1.

Thus the experiments must be able to distinguish among different groups, different goals for meetings, and different settings for the agents' parameters and models.

As a partial counterpoint to these difficulties, the system allows for Wizard of Oz experimentations, as explained in section 5. That means that it is not necessary to develop a complex agent to verify if it would be useful for a particular group/meeting type. A human, masquerading as an agent, can augment the meeting: the human is the one that does the interpretation of data, comparison with the normative theory, and deliberation on how and in what format to act, but the output is channelled to the system's devices. A Wizard might thus be responsible for figuring out that John needs encouragement to contribute to the meeting, decide that a public encouragement is the appropriate action, and generate the string "let's hear what John has to say" that is sent to the Kwabena voice generator. The participants would have the impression that it is Kwabena who is suggesting that.

In conclusion, this research provides an innovative approach to meetings: the augmentation of meetings by using tools, agents, and virtual participants. We have shown that it is possible to construct seemingly smart agents to augment meetings. We still have to evaluate in which conditions what agents are indeed useful to different types of meetings.

References

Bales, R. F. (1999). Social Interaction Systems: Theory and Measurement. *Transaction Publishers, New Jersey, USA.*

Bales, R. F. and Cohen, S. P. (1979). SYMLOG: A System for the Multiple Level Observation of Groups. *The Free Press, New York, USA.*

Barthelmess, P. and Ellis, C. (2002). The neem platform: an evolvable framework for perceptual collaborative applications. Technical report, University of Colorado at Boulder, Computer Science Department. http://www.cs.colorado.edu/˜skip/NeemPlatform.pdf.

Biddle, B. (1979). *Role Theory: Expectations, Identities, and Behaviors.* Academic Press.

Conklin, J. and Begeman, M. (1998). gIBIS: An hypertext tool for exploring policy discussion. In *Proceedings of the Second Conference on Computer Supported Cooperative Work.*

DeSanctis, G. and Gallupe, R. B. (1987). A Foundation for the Study of Group Decision Support Systems. *Management Science.*

Ellis, C., Begeman, M., et al. (1986). Meetings augmentation and analysis. In *Proceedings of the First MCC/ACM CSCW Conference.*

Ellis, C. A. and Wainer, J. (1999). Groupware and Computer Supported Cooperative Work. *Chapter 10 of "Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence",Editor G. Weiss, The MIT Press.*

Gordon, M. (1985). *How to Plan & Conduct a Successful Meeting*. Sterling Publishing Co.

Hare, S. E. and Hare, A. P., editors (1996). *SYMLOG field theory : organizational consultation, value differences, personality and social perception*. Praeger.

Hentschel, G. C. (1984). There's a method in the magic: The nominal group process,. *Lifelong Learning*, 8(3):11–14.

Isbister, K., Nass, C., Nakanishi, H., and Ishida, T. (2000). Helper agent: An assistant for human-human interaction in a virtual meeting space. In *Proceedings of the CHI 2000 Conference*.

Jebara, T., Ivanov, Y., Rahimi, A., and Pentland, A. (2000). Tracking conversational context for machine mediation of human discourse. In *AAAI Fall 2000 Symposium - Socially Intelligent Agents - The Human in the Loop*.

Linstone, H. and Turoff, M. (1975). *The Delphi Method: Techniques and Applications*. Addison-Wsley Publishing Co.

Nakanishi, H., Yoshida, C., Nishimura, T., and Ishida, T. (1996). Freewalk: supporting casual meetings in a network. In *Proceedings of the ACM 1996 conference on Computer supported cooperative work*.

Nishimoto, K., Sumi, Y., and Mase, K. (1998). Enhancement of creative aspects of a daily conversation with a topic development agent. In *Coordination Technology for Collaborative Applications – Organizations, Processes, and Agents*, volume 1364 of *Lecture Notes on Computer Science*, pages 63–76. Springer-Verlag.

Polley, R. B., Hare, A. P., and Stone, P. J. (1988). The SYMLOG practitioner: Applications of Small Group Research. *New York, Praeger*.

Robert, H. M. (1915). *Roberts Rules of Order Revised for Deliberative Assemblies*. Scott, Foresman, Chicago.

Wainer, J. and Braga, D. P. (2001). Symgroup: applying social agents in a group interaction system. In *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, pages 224 – 231.

Whitaker, R. (1994). GDSS' Formative Fundaments, an interpretive analysis. *Computer Supported Cooperative Work (CSCW)*.