Considerate Audio MEdiating Oracle (CAMEO): Improving Human-to-Human Communications in Conference Calls

Rahul Rajan,
Ted Selker
Carnegie Mellon University Silicon Valley, Mountain View, CA
Ted.selker@sv.cmu.edu

Carnegie Mellon Silicon Valley Report # 08/2013/010
Similar to publication in DIS 2012

ABSTRACT

This paper introduces CAMEO, a behavior-driven design approach to address commonly occurring technical and social problems in audio-only teleconference calls. Many of these problems are associated with the missing visual channel and the low bandwidth for non-verbal signals. CAMEO seeks not only to sense these problems, but also to frame and respond to them in considerate ways. These include scheduling of advisory feedback prompts, and assistive feedforward mechanisms to augment this bandwidth constrained medium. This paper describes their implementation in CAMEO using a blackboard architecture that shapes and define its behavior. Two experiments were conducted to evaluate CAMEO on its resolution of conversational dominance in a collaborative meeting, and its utility in reducing the effects of disruptive background noise on a conference call. The participants were asked to solve hangman and chess puzzles by collaborating on a multiparty conference call. We show that variance in conversational dominance can significantly be reduced with proactive aural feedback. Our experiments further reveal that such feedback can also reduce the impact of background noise on conversations. **Author Keywords**
Considerate, Facilitator, Multiparty, Audio, Conference call.

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: User Interfaces— Interaction styles; H.5.3.Information interfaces and presentationGroup and Organization Interfaces—Computer-supported cooperative work

General Terms
Design, Human Factors

INTRODUCTION
Globalization and technology have brought radical changes to business practices, and meetings in particular. With increasing frequency, teams are being composed of members from Submitted for review to DIS 2012. geographically different locations so that they can bring to bear their expertise on pressing problems, without the travel and associated costs. These distributed teams collaborate by holding meetings on conference calls and other networking solutions. By the very nature of the distributed setting, a host of technical, organizational and social challenges are introduced into these meetings that have been well documented and studied [33, 19]. A number of these challenges are associated with the missing or attenuated channels of non-verbal communication which affects basic interaction constructs such as turn-taking, speaker selection, interruptions, overlaps and backchannels [16]. In this work, we explore how technology can aid communication by better accommodation for these social signals, and by creating new ones. We focus on the group communications in a distributed audio-only conference call, where the participants are collaborators in a problem-solving/ decision-making meeting. Audio conferencing ranks only behind telephone, fax and email in terms of most used collaboration technologies. [24].

In a collaborative setting, teams with constructive interaction styles achieve better performance (e.g., solution quality, solution acceptance, cohesion) than teams with passive/defensive interaction styles [25]. Team interaction styles are a reflection of the aggregate communication traits of the individual members. Higher variations in extraversion between team members lead to less constructive and more passive/defensive interaction styles within teams [2]. Shared leadership is a critical factor that can improve team performance [8]. This leads us to ask if it is possible to influence the group dynamics by encouraging extraverted people to share the floor when their dominance is pronounced. We might hypothesize that this would make the interaction style of the group more constructive, and lead to higher satisfaction and performance.

Indeed, it has been shown that providing feedback about the group dynamics helps participants modify their behaviors [30]. If this is the case, what types of feedback and feedforward mechanisms can computing systems employ to tackle other behavioral problems prevalent in audio teleconferencing. Taking it a step further, it sets us up to explore ways a computing system can abet people social skills, like it does their cognitive skills. Auto-correction features in text processing applications show how representation of accepted grammar have allowed computers to fix syntactic disfluencies; we seek to recognize social disfluencies between people and use computers to improve their communication. To explore the idea of an agent proactively interjecting social feedback on an 1 audio channel, we built CAMEO (Considerate Audio MEeting Oracle), a multiparty conference call facilitator.

This paper is organized as follows: after a survey of related work, we discuss the design goals and features of CAMEO. We then describe the architecture and implementation details of a test bed system that facilitates audio conference calls between two or more people. We show how this audio interface can help distinguish among the participants in a meeting, make people aware of their behavior on the meeting, and deflect common interruptions. We present results from preliminary evaluations of two of these features, namely, the resolution of conversational dominance and disruptive background noise. We also discuss insights gained while designing and experimenting with the other features of this system.

RELATED WORK

Many researchers have tried to overcome the shortcomings of distributed collaboration. Erickson and Kellogg formulated the concept of social translucence [13] to facilitate fluid and productive online group interactions. Their ideas informed the design decisions in a lot of ensuing work on group conferencing solutions. They advocate that the three properties a socially translucent system must posses are visibility, awareness, and accountability. Visibility and awareness brings about a collective awareness creating an environment where individuals feel accountable and responsible for resolving problems. Together they form the building blocks of social interaction, and allow mechanisms for social control like norms, rules and customs to play out in a distributed setting.

These ideas were employed by Yankelovich, et al., in the design of their Meeting Central system to address the problems with audio conferencing which were documented in a series of studies [33]. They grouped the problems into three categories: audio, behavior and technical. The top problems that affected meeting effectiveness included too much extraneous noise (audio), and difficulty in identifying who was speaking (technical). Among the other reported problems, participants had difficulty knowing who had joined or left the meeting (technical), and speakers not realizing that they were not close enough to their microphones (behavior). More interestingly, the authors note that “most audio problems are, in fact, behavioral. They are compounded by the difficulty remote participants have, both technically and socially, in interrupting to indicate that the problem exist” [33].

The idea of using feedback to influence group dynamics and behavior in distributed meetings was further explored by Kim, et. al. in [22], where they focused primarily on the effects of dominance. Their Meeting Mediator system computes group interactivity and speaker participation levels, and uses a visualization to feed this information back to the participants on their mobile phones. They showed that dominant people had a negative effect on brainstorming as fewer ideas were generated during these sessions. They also found that dominant people caused more speech overlaps in distributed meetings. Since spoken communications are so dynamic, the question arises as to how facilitation can be achieved at the turn-taking level, to manage these interrupts or overlaps. The old solution of contribution minders in Robert’s Rules of Order or the timers in debate formats might be made more fluid. A modern version of imposing control on a conversation has taken the form of cartoon characters called embodied agents that are virtual participants.

The success of embodied conversation agents depends on advanced behaviors to the situational context. The focus has been on endowing these systems with facilities to communicate and respond through the production of language and associated non-verbal behavior (gaze, facial expression, gesture, body posture) [3]. An early and commonly used actionselection approach is the Do the Right Thing architecture that provides the ability to transition smoothly from deliberative, planned behavior to opportunistic, reactive behavior in order to maximize task efficiency while maintaining trust [23]. Similarly, multiparty dialog systems attempt to make turn-taking and other conversational dynamics more fluid to avoid communication breakdowns [7, 6].

There has also been an effort towards long-term behavior adaptation through the use of emotion and memory. [15] describes a reflective architecture for agents where detection of emotional stress or frustration can trigger re-evaluation of past behavior, and the setting of new strategies and goals. They showed that such adaption can extend the range and increase the behavioral sophistication of the agent without the need for authoring additional hand-crafted behaviors. [12] describe an affect-sensitive Intelligent Tutoring System that models and responds to students' affective states in addition to their cognitive states. A vast majority of ubiquitous systems, however, don't have embodied agents, but their response remains an opportunity for improvement. [34] describes cognitive user interfaces that try to respond appropriately to error-prone user input like gestures. However, it does not focus on the social aspects of interaction.

The work discussed so far used GUIs and visualizations. Graphical interfaces to computers were developed over the last few decades as a high bandwidth parallel communication channel. A computer interface can change the look of any part of a screen at any place in a fraction of a second, while the eye can notice millions of stimuli simultaneously. A keyboard allows directly coded input to control the computer, while a mouse or touch screen allow a person to react to concrete interface items directly. An audio interface has none of the afore-stated advantages. All information is layered on a low bandwidth interface and without a keypad there is no direct manipulation. Indeed, introducing an agent into an audio environment implies that they must successfully cohabitate the same environment as the participants. However, work done by Rienks, et al., reveal that participants found voice and visual feedback to be equally efficient [26]. While voice messages block audio and so were more intrusive than text messages, participants of the meeting appeared to be much more aware of their own behavior when the system provided vocal feedback. They also reported that as they got used to the interface they found it less disruptive.

CONSIDERATE AUDIO MEDIATING ORACLE (CAMEO)

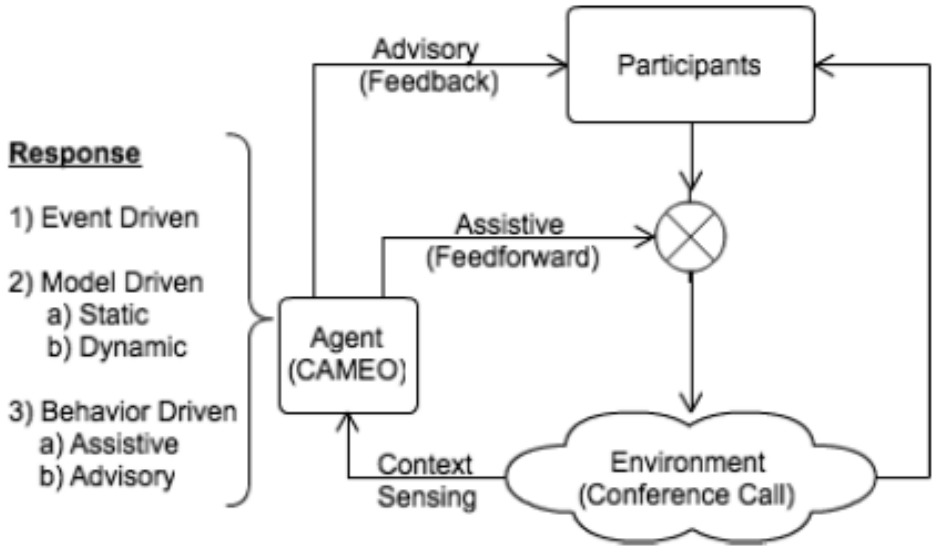

Figure 1: Considerate response using assistive feed-forward and advisory feedback approaches.

We use the term "considerate" to refer to the unstated norms and mechanisms of social control that people engage in while communicating with each other [29]. It is analogous to a control system used in any dynamical system — to

obtain a desired outcome the different inputs to the system need to be manipulated and regulated. As a participant of the meeting (a dynamical system), our CAMEO systems strives to becomes a useful social actor and will have to engage in the same forms of social control, as according to Weber, “An action is social insofar as its subjective meaning takes account of the behaviors of others and is thereby oriented in its course [32].”

On the other hand, many intelligent interfaces take the simplest approach to improving communications by varying the mapping from the inputs to the outputs. We see an adaptive component between the input and output, such as the one found in COACH [28] or modern adaptive interfaces, that rely on explicit models of context as a major change from the classic sense and react paradigm. We build on this by adding a social model of the other, and how they perceive CAMEO, that shapes CAMEO’s behavior. In order to successfully engage with the other participants and positively influence the meeting, we distinguish between CAMEO’s assistive feed-forward behavior that modifies the channel to directly effect a change, from the more subtle advisory feedback behavior that encourages users to effect a change (Figure 1). Feed-forward is when CAMEO tries to influence the meeting through its own actions. For example, when there is a prolonged overlap between two participants, CAMEO could delay or pitch shift one so that the others can make sense of what is being said. Feedback on the other hand relies on the user, allowing them to monitor themselves and self-correct. For instance, people might not realize how loud they sound to others on the phone. But if they could hear themselves, or if CAMEO could hint at this before someone on the line does so, it might save the embarrassment and any disfluencies in communication.

CAMEO’s behavior is also governed by its goals which are twofold — to increase intelligibility and to improve sociability. By intelligibility we mean how comprehensible speech is, and how easy it is for the information contained in the speech to be understood. Sociability is about the social signals that normally accompany the delivery of informational content during communication, and its import on the social dynamics. In so far as these two dovetail, it is worth appreciating that the solution to one cannot be completely divorced from the other. While someone might be speaking so softly as to not be intelligible to others on the line, their reason for doing so might be a social one. Perhaps they are shy, or by speaking softly they could be trying to get the others to calm down and pay attention. Simply normalizing the volume in this case to increase intelligibility might affect the intended social outcome. Thus in our design, the constraint is on giving bandwidth to productive social signals that might be lost in audio-only communications, and on creating new ones to augment communications. In this way the system aims to increase intelligibility while improving sociability. CAMEO’s approach to solving some of the often cited problems of audio teleconference calls illustrate this, and are described below.

Dominance Detection

A Dominator is a type of self-oriented behavioral role that group members can occasionally slip into during a meeting[20]. Groups dominated by individuals performing these roles are likely to be ineffective [18]. On the other hand, Dominators also drive discussions and generate consensus [31]. Thus, it seems that while too much dominance might stifle contributions from the other participants, too little can reduce consensus and decision making because of a lack of a clear social order [1]. This opens up room for considerate feedback to the Dominator, where simply telling them or showing them that they are being dominant [31, 10] might not be as effective as encouraging them to use their status more positively at the appropriate times. In our current implementation, once CAMEO has detected that someone is dominating the meeting, it says subtly says “turn taking?” on the dominant person’s channel alone. If it is close to the end of the meeting, it might mention how much time is left to encourage them to consider leaving space for other participants to contribute. If they are raising their voice, it can artificially make their voice even louder in their earpiece. The aim is to make the user cognizant of their behavior in the hope that they reflect to self-correct without being intrusive and demeaning. Similarly if someone is being dormant, CAMEO will say “any thoughts?” to encourage their participation. In both cases the feedback has its purpose embedded within it and aspires to be natural. Spatial techniques could be used to be less intrusive, for example using the left/right stereo channels to play the notification on one while the group communications continues to be heard on the other.

Background Noise Detection

The average values of the non-speech audio buffers can also be used to determine if the background noise is too loud. Sometimes a meeting participant might be in a noisy cafe, or in a moving vehicle and while they might be able to tune out extraneous noise sources, it can be harder for the participants on the other end of the line to ignore it. The nature of the meeting or the roles of the different participants can sometimes make it inconvenient for the others to point this disturbance out. More crucially, it is hard for the participants to pinpoint whose channel is responsible for introducing the noise, and the process is cause for potential embarrassment. 3 To pre-empt this CAMEO tries to

detect high levels of background noise and discreetly provides feedback to the offending participant by letting them know its “noisy”.

Volume Meter

CAMEO notifies the speaker of an improper speaking or microphone volume by comparing signal energy and the noise floor to threshold values that were set experimentally. As long as the signal volume lies within an optimal range determined empirically, the speaker should be heard clearly. Speech and non-speech are distinguished by looking at the log energies of audio buffers. Speech buffers are used to determine signal values, while non-speech buffers are used to determine noise floor values. During a meeting, if the signal volume is too high, it could be that either the microphone is too loud or that the speaker is speaking too loudly. This is determined by calculating the signal-to-noise ratios. If the ratio is high, then most likely the speaker is loud, and if the ratio is low then most likely the microphone volume is up. Similarly, if the signal volume is too low, a low noise floor indicates that the microphone volume is too soft. If the noise floor is to high, the speaker is being soft and is encouraged to speak up.

Speaker Identification & Presence

Current audio teleconferencing systems can involve many participants who may be speaking simultaneously, joining and leaving during a conversation, and unwittingly speaking over other participants in a conversation. Though some of these issues are caused by noise or missing in-person cues, unique background noises might orient listeners. For example, if Ron is playing music in the background, or Joe is driving, it becomes immediately obvious when either of them go offline. We are subconsciously aware of their presence on the line even when they are not speaking. Our system can play background tracks like music, tones, or ambient noise to annotate and indicate presence of participants during a conversation. For instance, a distant orchestra adds a new instrument whenever a person enters a conversation and ceases playing the instrument when that person leaves the conversation. While these background tracks might be distracting, we want to augment the voice signatures of unfamiliar participants in the hope of making it easier for the group members to identify each other when one speaks.

Entry & Exit

As noted in [33], it can be hard to tell when participants get dropped from or reenter a conference call. Existing conference call systems can commit considerable time to loudly announcing entry and can be annoying. We look at more considerate ways to notify the other participants of these events, including detecting when the meeting floor is free and using intonations. We want to convey the most information in as small an audio footprint as possible

SYSTEM ARCHITECTURE & IMPLEMENTATION

The Backbone: CLAM & JACK CAMEO is built on CLAM (C++ Library for Audio and Music), a framework for audio processing, along with the JACK Audio Connection Kit on Ubuntu Linux. Other tools including PortAudio, SoundTouch, and Audacity were considered but CLAM was chosen for its robust processing feature set and modular API. JACK is used as the audio server for its real-time network streaming capabilities. The audio inputs are routed through JACK into a CLAM network, which runs each meeting. The two most significant components of CAMEO are the Channelizer, which represents a speaker in the conversation, and the Supervisor, which represents the meeting facilitator.

Channelizer

Each speaker is represented by a Channelizer object in the system, which is local in scope and only concerned with the individual speaker. It consists of four main components:

The first component handles all low-level energy/buffer/sliding window calculations, which deals with the raw input data provided by the microphone and the CLAM framework. Our testing scenarios include up to four speakers, although this number can scale up to any desired amount. The microphone inputs are sampled at 16kHz each, with 64 frames generated per period and 2 periods in a buffer. These 8 ms buffers are pushed into a sliding window that is 30 buffers long, with a window step size of one buffer, i.e. there is no overlap. The estimated system latency is 8 ms.

The second component determines the “Participant State” given the current energy level in each sliding window. For every buffer, the peak sample is found and averaged over the sliding window to determine if the participant is speaking. This allows CAMEO to adequately identify samples with speech and samples with no speech.

The third component transmits each speaker's statistics to files and an external Rails server for data visualization to help researchers gain insight. The user action and feedback model is transmitted asynchronously via XML-RPC to the Rails site, which runs a publish-subscribe service for all connected clients. This allows the researchers to see real-time updates on various meeting statistics, changing social dynamics and how CAMEO is affecting each of these.

The last component generates alerts for various speaker-specific events, which are sent to the user through a text-to-speech module. These include alerting the speaker if he is speaking too loud, too soft, or if the background ambient sound is too loud. The alerts generated here are buffered in an internal queue specific to each Channelizer before being pushed out to the meetings main priority queue (Figure 2). This is CAMEO's blackboard allowing different models to be used for determining priority among different kinds of alerts, all of which are user specific. This facilitates working with different knowledge sources, user profiles and meeting types in future experiments (e.g. CEO, guest speaker, instructional meeting, etc.).

Participant States In order to mediate the conversation, the system allows each participant to be in one of four states:

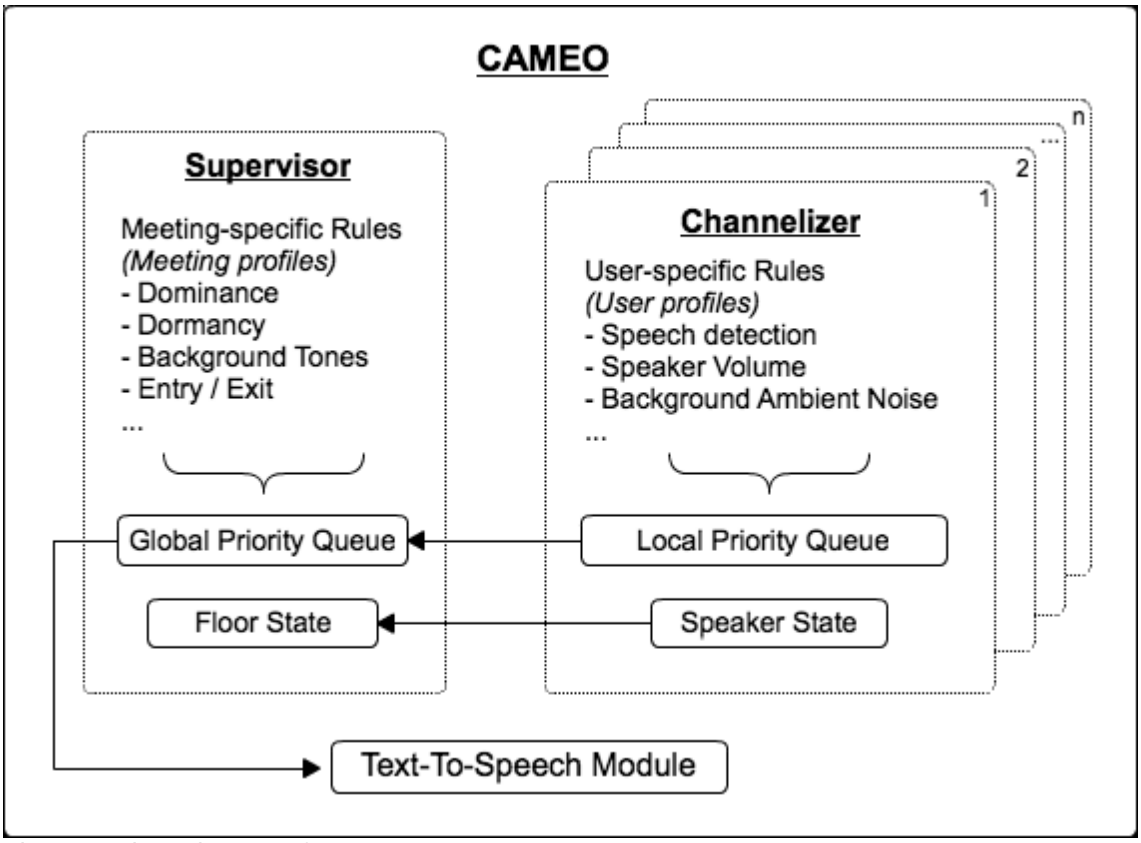

Figure 2: Flow Diagram of CAMEO.

_ Not Talking: The participant is silent and does not have the floor
_ Start Talking: The participant begins to talk and wants the floor
_ Still Talking: The participant is still talking, and if he does not have the floor, still wants it
_ Stop Talking: The participant is no longer talking, and if he has the floor, relinquishes it

These states are combined with the floor control model described below to detect audio cues for the Supervisor to act upon.

Supervisor

The Supervisor is global in scope, monitoring the Channelizer objects and provides the capability to decide and act on the social dynamics between the participants. The Supervisor consists of four knowledge sources.

The first component calculates each speaker's dominance by calculating how active each person is relative to activity level of the other participants. The Supervisor calculates each participant's dominance with the following equation:

$$Dominance_{P_x} = TSL_{P_x} / \sum_{i=1}^{n} TSL_{P_i}$$

TSL is the Total Speaking Length of a particular participant. This dominance measure is useful in resolving conversational conflicts such as interruptions, as well as monitoring how effective CAMEO is in fostering collaboration across all participants (see Figure 3).

The second component is the global priority queue that employs a blackboard architecture coordinating multiple knowledge sources [14]. It decides how to communicate the responses from the different knowledge source based on its considerate response goals. For example, it combines similar messages that occur consecutively. It reorders or delays messages based on their importance, the time since the last alert and the number of alerts. It can also choose to announce messages based on the floor state. For example, while entry and exit events have the highest priority, it might wait for the floor to be empty before making an announcement. The various knowledge sources also impose considerate response goals. The dominance knowledge source, for example, only makes a request when there is a conflict for the floor (interrupts) between a dominant person and dormant person. The background noise knowledge source operates on a reinforcement scheduler to limit the annoyance of its responses.

This mechanism also allows for different models for determining priority among alerts, similar to each Channelizer's internal queue. For example, in a collaborative scenario, CAMEO will give higher preference to dormant participants, whereas if CAMEO was setup to facilitate an instructional scenario CAMEO will give higher preference to the instructor. This flexibility in reasoning that the architecture allows us will be useful in adding and testing more considerate features in the future.

The third component detects and resolves any conversational collisions based on which person is more dominant or dormant. In order to encourage collaboration across every participant, CAMEO favors dormant participants in the event of a verbal collision or interruption. When such an interruption occurs, the system generates a special alert and sends it to the meetings main priority queue for processing. This allows for adding and experimenting with different meeting models, which will change which participants we favor in a given interruption.

The fourth component translates a speaker's state into a "Floor Action", and then determines which channel has the Floor. This mapping from a speaker's action to floor ownership will determine which channel is more entitled to speak during a verbal collision or interruption. By translating a speaker's action to the meeting floor state, we are also able to determine which speaker is more dominant.

The last component processes any alerts in the global priority queue, and notifies all specified channels via text-to-speech. Our system is able to alert users with Festival, a C++ speech synthesis library. Messages are either broadcast to the entire meeting (such as entry and exit), or only played on a specific channel (i.e., "You are speaking too loud").

Floor Actions

In order to facilitate the conversation, the system uses the idea of a meeting floor, similar to what was done in [5]. Each participant has a floor action object, which can be in one of four states:

- No Floor: The participant is not speaking
- Take Floor: The participant starts to speak
- Hold Floor: The participant is still speaking
- Release Floor: The participant is done speaking

Floor Control

This identifies which user currently has the floor. The floor can only be taken by another participant when the floor owner releases it. This model allows the Supervisor to detect and measure audio cues easily in order to identify what actions CAMEO should take to socially enhance the conversation.

Audio Cues

The Supervisor picks up several different audio cues which have been proven effective in distinguishing the speaker context of a conversation [21]. The cues focused on are:

- Total Speaking Length (TSL) The amount of time a person speaks over the course of the entire conversation.
- Total Speaking Turns (TST) The number of distinct times a person speaks over the course of the entire conversation.
- Total Speaking Turns without Short Utterances (TSTwSU) The number of distinct times a person speaks, not including any short utterances or affirmations made.
- Total Successful Interruptions (TSI) The number of times a person has successfully interrupted another person. The number of times a person interrupts another is an indication of dominance.

[21] showed that using a combination of these cues to classify conversational dominance yielded an 88.2% accuracy on a fairly typical meeting corpus, which is why we chose the above metrics.

Actuators

Currently CAMEO has a number of actuators. It controls who talks to whom, and uses this to implement floor control. It can change the amplitude and frequency of the input channels and can mute or delay them. It can also overlay background sound or introduce reverb and other effects. It does all of this for any combination of the participants. For example, it can introduce a feedback that only participants two and three can hear, or it can delay participant one's speech to participant three.

PRELIMINARY EXPERIMENTATION

To begin evaluating the system we tested a couple of features separately. Many of the initial implementations of these ideas about supporting conference calls with CAMEO were hugely disruptive. As soon as CAMEO began entering a conversation, it became viscerally obvious how easy it was to disrupt it. The idea of identifying entrances and exits was explored in various ways that were as distracting as the "¡participant¿ is entering the meeting" pronouncement used by most teleconferencing solutions today. To identify people with background sound annotation, we tried adding an instrument track in the background for each person. First the music was distracting, second it was difficult to remember any mapping between individual and instrument. Generic background noises of flowing river, an office setting, and traffic were tried to similar deleterious effect. Three dimensional audio in which people were positioned in space has seemed too vague to be useful yet. We ended up using synthesized tones of a marimba at different frequencies (C4, E4, G4) to annotate the different speakers. It is possible that the presence indicators could be made more subtle and effective but experiments with the feedback approaches for dominance detection and background noise detection bore the first success. The setup for each experiment is described in further detail below.

For dominance detection, we initially started off with CAMEO prompting users with the message, "You have been talking for a while. Please give others a turn". It turned out that while this was acceptable the first time, every time after that it became less and less tolerable. It was not just that in a meeting participants have low cognitive bandwidth for a third-party. It was also "nagging", as one of the participants put it. Changing the message to, "take turns" (a directive) or "turn-taking?" (a suggestion), allowed the agent to be subtle.

In our study, we attempted to get CAMEO to decrease the difference between dominant and non-dominant people, i.e. to lower the variance in dominance as the meeting progresses, without disrupting the flow noticeably. We also sought to have CAMEO successfully encourage more interactivity, i.e. the turn-taking will be more balanced, and the speech utterances of all participants will be shorter on average. In the background noise detection case, we attempted to detect background noise and provide feedback about it to reduce disruptions caused by the noise. We hypothesize that the meeting will run smoother with participants interrupting each other fewer times.

Evaluation

Dominance Detection

Our first set of experiments evaluated the dominance detector component implemented in CAMEO. We conducted a study with twelve groups of three participants each. The groups were formed by drawing from a pool of nineteen volunteers. The participants were students and research scientists (5 females & 14 males) belonging to the same campus, with the youngest being 21, and oldest 43. The collaborative behavior of people with different partners is dramatically different. Indeed, group interaction styles reflect aggregation of communication traits of its team members [2]. Even though a participant took part in multiple groups, all of the 12 groups had unique compositions drawing different dynamics from the 19 participants. On half of the groups we ran the control condition first (CAMEO Off), while on the other half we ran the test condition first (CAMEO On).

We performed a within-subject experiment comparing how the groups behaved with and without CAMEO. The participants were located in physically different locations with computer terminals that had screen sharing and control enabled. Each group went through two problem-solving sessions, one with CAMEO and one without, for a total of twenty-four sessions. The sessions were held back-to-back and were five minutes- long each. During each session, the participants collaborated on playing Hangman, a word guessing game. The game and its duration were chosen so as to simulate a slice of an actual meeting where everyone is an equal collaborator, and a higher group extraversion would be beneficial to 6 (a) (b)

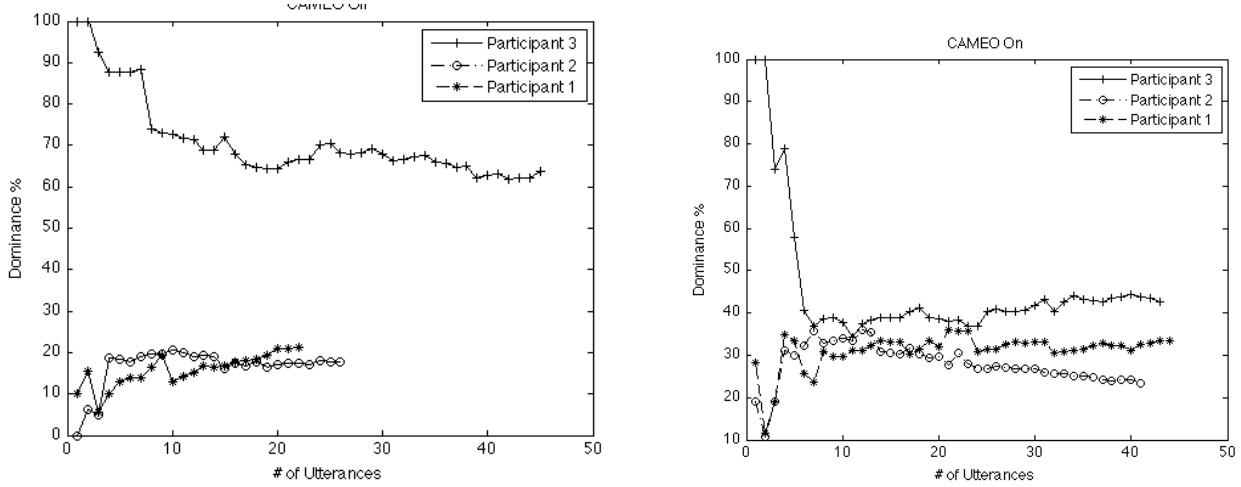

Figure 3: Dominance (%) vs. Number of Utterances during the Dominance Resolution evaluation for one of the test groups: With CAMEO On (b), the Speaker 3 becomes less dominant and Speakers 1 and 2 become less dormant. Also, they contribute more equally, i.e., the number of utterances from each is around the same, with CAMEO On.

the groups performance. The protocol was that the three participants would agree on a letter before entering it; the last person to agree would input the letter into the terminal. For the purposes of this experiment we measure a participant's dominance level as a fraction of their Total Speaking Length (TSL) divided by the TSL of all participants which was also shown to be a reasonable measure of dominance in [21]. The dominance percentage threshold we use is 40%. The prompt is only activated when there is a turn-taking conflict after this threshold has been reached. A turn-taking conflict is a speech overlap between the dominant user and another user that is longer than one second. These values were heuristically determined to work well.

For interactivity we calculate Turn-taking, which is the ratio of the TSTwSU (Total Speaking Turns without Short Utterances) of the most dominant person to the TSTwSU of the least dominant person. TSTwSU includes only utterances that were longer than simple feedback like "umm" or "yea". Turn taking ratio gives us a measure of how well the floor was being shared between the participants. Turn-taking ratio of one,

Background Noise Detection

A second scenario was created in CAMEO to ameliorate background noise disruptions. This was tested in another set of experiments with three groups of three participants each. The groups were formed by drawing from a pool of seven volunteers. They were all male, with the youngest being 21, and oldest 28. We performed a within-subject experiment comparing how the groups behaved with (experimental condition) and without CAMEO (control condition).

The participants were located in physically different locations with computer terminals that had screen sharing and control enabled. Each group went through six problem-solving sessions, three with CAMEO and three without, alternatively, for a total of eighteen sessions. The sessions were held back-to-back and were four-minutes-long each. During each session, the participants collaborated in discussions around solving chess-puzzles presented on their screen, of the mate-in one/two/three variety. The game and its duration were chosen so as to simulate a slice of the meeting where the cognitive load on the participants is high, requiring concentration and memory. The protocol included that the three participants would agree on a move before executing it; the last person to agree would input the letter into the terminal.

The participants were instructed to respond naturally as they would if the background noise on a telephone line was too loud, and that if it was disrupting the meeting they should ask for it to be turned down. At different intervals in the game, a TV program would be played close to one of the terminals to introduce background noise into the meeting. If the participant on that terminal was prompted to reduce the volume either by CAMEO or by one of the participants, they would do so by pressing a button on the provided MacBook remote control. After an interval of thirty seconds to a minute, the background noise would be introduced again. CAMEO is built to prompt on a reinforcement schedule, i.e. subsequent prompts would be further and further apart, unless a sufficient amount of time had lapsed since the last prompt. This was done to model a thoughtful human response to repeated occurrence of background noise on the channel, as opposed to setting off a prompt every time background noise was detected.

To study the effect of background noise on the flow of the meeting, we use the Total Successful Interruptions (TSI) metric, i.e. the number of times a participant successfully interrupts another.

Results

Dominance Detection

CAMEO had a strong effect on the dominance levels. As the meeting progressed, dominant people became less dominant, 7 and dormant people took the floor more often. To quantify these results, we calculated the variances of the dominance levels of all participants at the one minute mark and at the end of the meeting, across all groups with CAMEO On and CAMEO Off (Table 1). The table shows that the meetings start with similar variance in dominance between the participant. At the end of the meeting, there is a bigger and statistically significant drop in variance with the CAMEO On, than with the CAMEO Off (F-test of 0.001 for dominance variance at the end of meetings with CAMEO On and Off).

	1 min.	End
CAMEO Off	23.07	13.31
CAMEO On	20.87	7.50

Table 1: Variance in dominance levels of all participants across all groups one minute into the meeting and at the end of the meeting.

CAMEO seemed to have a positive effect on interactivity. It appears that the most dominant person was taking the floor less when CAMEO was facilitating the meeting, but there was not enough experimental data to show statistical significance (mean Floor Taking ratio = 1.84 with CAMEO On, compared to 2.51 with CAMEO Off, two-sample T-test: $p = 0.12$, Figure 4). There was not much difference in the average speech utterance of the participants (mean = 1.63 with CAMEO On and mean = 1.68 with CAMEO Off, two-sample T-test: $p = 0.37$, Figure 4).

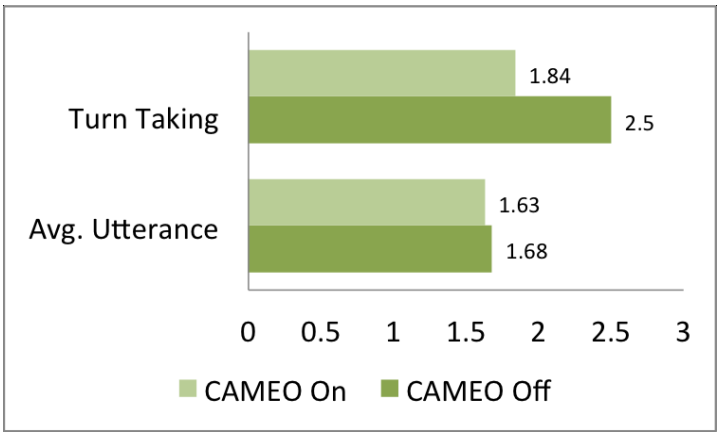

Figure 4: Floor Taking Ratio and Average Speech Utterance with CAMEO On and Off. Means=(1.63,1.68,1.84,2.51), SE=(0.07,0.08,0.23,0.53)

Background Noise Detection

CAMEO *was able to positively impact the flow of the meeting*. With CAMEO On, the average number of times any participant interrupted another in a four minute session was reduced by almost half, and was shown to be statistically significant (mean = 7.05 with CAMEO On, and mean = 12.44 with CAMEO Off, two-sample T-test: p = 0.007, 2).

	CAMEO On	CAMEO Off
TSI	7.05	12.44

Table 2: Average number of interrupts (TSI) among participants solving chess-puzzles in four-minute sessions.

Discussion

CAMEO is an audio-only solution to tackling some of the problems in conference calls, and presents a departure from the more popular methods of using peripheral visual interfaces. We argue that the advantages to this are two-fold. Firstly, the participants can focus on their tasks better, without having to continually interpret feedback from a peripheral interface. Audio feedback is just-in-time and unambiguous in its intent. Secondly, advisory feedback is private. Communications between a participant and CAMEO occurs unbeknownst to others, creating a different dynamic compared to when the information is displayed to all participants [22, 11].

From our observations, participants were not intentionally being dominant, or negligent of background noise. For the most part they seemed so engaged in their task that they were not actively conscious of their behavior. We believe that making them so might be an added cognitive and social load. Instead, by prompting them at just the right time we can reduce this load, but it needs to be done considerately though. CAMEO feedback is now subtle enough that most users claimed not to pay attention to the prompts while engaged in the task of playing Hangman. It is here the system makes its strongest statement. The data clearly shows that the system made a big impact on their actions. This is an important and novel finding that points to the utility of pro-active feedback approaches in the audio domain that gently nudges users towards desired behaviors.

Evaluating the background detection feature brought out the most interesting observations. When background noise was introduced into a conference call which had so far only had the audio from participant voices, a participant would invariably ask for it to be reduced. However, they would be a lot more tolerant to subsequent introduction of the background noise. Only when the noise was increased to a volume higher than before would they point out the disturbance. Also what we observed was that the more absorbed they were in the discussion, the less they noticed the background noise. In fact, most participants stated that they were not affected by the noise. However, it was obvious to a third party observer that the noise was actually affecting the meeting. Participants would talk louder to counter the background noise. As they raised their voice, they also became more aggressive in wanting to get their point across. Clearly, the background noise was introducing stress into the meeting, although after the sessions, the participants themselves would deny it. With CAMEO on, its prompts precluded the background noise from affecting the meeting to such an extent.

We also believe that the system did not negatively affect performance between the test and control case. Indeed, during the 5-minute experiments, the number of words attempted the number of wins and loses, and the number of wrong letter guesses was not distinguishable between the test and control conditions. The goal of this work was to test whether feedback on a unimodal communication channel positively influences behavior. We hope our results of reduced dominance in even a short cooperative collaboration will help focus future 8 work. We also expect that social interplay and performance will be linked, and with some interface improvements, systems might have other properties such as relationship building, maintenance, etc., beyond simple group performance.

Future Work

The models we used for dominance are static which served the purpose of this study. Dynamic models, however, can more accurately predict when a group member is negatively affecting a meeting by looking at other cues for dominance like interrupts and overlaps, and even for patterns in these cues over time. However, detecting the intention of interruptions and overlaps are not straightforward [27]. The participants could be co-constructing a thought as often happens, or could be confirming each other through repetition. Interruptions and overlaps in these cases are not attempts to take the floor, although they might be construed as acts of dominance. Instead, looking for patterns of repeated acts of dominance might allow CAMEO to infer a state of dominance and respond more appropriately. A state of dominance could potentially be modeled statistically using a Markov process or n-gram model. Furthermore, by predicting the roles of the participants, the relationships between them and the type of meeting they are in, it should also be possible to set different interactions parameters to improve facilitation by a virtual meeting facilitator.

For speaker presence and identification, we further experimented with automatic annotation by replacing the background orchestra tracks that were assigned to each user with tones of a chord progression. We initially had them sound at the end of an utterance. In a two-person setting, its effect was to subtly moderate turn-taking. Participants paused a little longer for the sound to play. The tone provided some sort of affirming feedback to the speaker that they had been heard. This was not the case when more than two people were on the line. We then moved the tones to the beginning of each speakers turn. Over a prolonged period this creates a musical pattern that reflects the group dynamic. We expect to study this as a subliminal audio feedback mechanism.

With the responses to entry/exit events and microphone/speaker volume detection features thrown in, what we learned is that even with such a small feature set, the nature of the prompts and their timing become crucial. We varied the nature of the messages by experimenting with word choice and length, intonations, and other effects. To experiment with the timing we implemented separate priority queues for each participant, and one global queue for the whole meeting. For instance, we might want to delay messages to a participant if they are too close together, unless a particular message has a very high priority. Quantifying and hard-coding ‘too close’ or ‘high priority’ can make the system fragile as humans are highly sensitive to the situational context, like in the case where the participants tolerance to background noise increased. To be successful, CAMEO and other proactive agents will also need to display better awareness to the situational context by responding in kind [17].

CONCLUSION

Intelligent interfaces can focus on various aspects of knowledge, including context and models for user, task, and system. New work also focuses on recognizing and modifying system understanding of users based on their affective stance. We present yet another paradigm of feedback which uses intelligence to react with considerate and socially motivated responses to a user. In specific, we challenged ourselves to create an intelligent system which collaborates using the same communication channel and modality as humans. We demonstrate the valuable influence a pro-active agent can have in a decision-making meeting. Our experiments focused on group dynamics in a collaborative Hangman game facilitated by CAMEO.

CAMEO incorporates principles of feedback at a variety of levels to succeed at reducing dominance and the impact of background noise. These included developing a language of short, unobtrusive nonjudgmental responses. It also includes a blackboard system to prioritize and arbitrate between knowledge sources and to schedule feedback. The scheduling aims to reduce interruptions through awareness of total message volume and message repletion. Our study showed that an advisory feedback mechanism (e.g. turn-taking suggestions) decreases the variance in dominance between participants. These mechanisms were also able to decrease the number of times participants interrupted each other due to background noise. Through these two studies of multiparty decision-making

collaborations, the paper demonstrates how CAMEO improves communication between the participants with statistical significance. Furthermore, it exposes the potential for such interfaces to positively influence human behavior in other computer-mediated domains.

REFERENCES

1. Bales, R. F. Social interaction systems: theory and measurement. Transaction Publishers, 2001.
2. Balthazard, P., Potter, R. E., and Warren, J. Expertise, extraversion and group interaction styles as performance indicators in virtual teams: how do perceptions of IT's performance get formed? SIGMIS Database 35, 1 (Feb. 2004), 41–64.
3. Bickmore, T., and Cassell, J. Social Dialongue with Embodied Conversational Agents. In Advances in Natural Multimodal Dialogue Systems, J. C. J. Kuppevelt, L. Dybkjæ r, and N. O. Bernsen, Eds., vol. 30 of Text, Speech and Language Technology. Springer Netherlands, 2005, 23–54.
4. Blattner, M., Sumikawa, D., and Greenberg, R. Earcons and Icons: Their Structure and Common Design Principles. Human-Computer Interaction 4, 1 (Mar. 1989), 11–44.
5. Bohus, D., and Horvitz, E. Computational Models for Multiparty Turn-Taking. Tech. rep., Microsoft Research, 2010.
6. Bohus, D., and Horvitz, E. Facilitating multiparty dialog with gaze, gesture, and speech. In International Conference on Multimodal Interfaces and the Workshop 9 on Machine Learning for Multimodal Interaction, ICMI-MLMI '10, ACM (New York, NY, USA, 2010). 7. Bohus, D., and Rudnicky, A. I. The RavenClaw dialog management framework: Architecture and systems. Computer Speech Language 23, 3 (2009), 332–361.
8. Carson, J. B., Tesluk, P. E., and Marrone, J. A. Shared Leadership in Teams: An Investigation of Antecedent Conditions and Performance. Academy of Management Journal 50, 5 (Oct. 2007), 1217–1234.
9. de Vreede, G. J., Vogel, D., Kolfschoten, G., and Wien, J. Fifteen years of GSS in the field: a comparison across time and national boundaries. In System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on (2003), 9 pp.
10. DiMicco, J. M. Changing Small Group Interaction through Visual Reflections of Social Behavior. Phd thesis, Massachusetts Institue of Technology, 2005.
11. DiMicco, J. M., Pandolfo, A., and Bender, W. Influencing group participation with a shared display. In Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW '04, ACM Press (New York, New York, USA, Nov. 2004), 614.
12. DMello, S., Lehman, B., and Graesser, A. A motivationally supportive affect-sensitive autotutor. New Perspectives on Affect and Learning Technologies (2011), 113–126.
13. Erickson, T., and Kellogg, W. A. Social translucence: an approach to designing systems that support social processes. ACM Trans. Comput.-Hum. Interact. 7, 1 (Mar. 2000), 59–83.
14. Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R. The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. ACM Computing Surveys 12, 2 (June 1980), 213–253.
15. Francis, A. G., Mehta, M., and Ram, A. Emotional Memory and Adaptive Personalities. In Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence. 2009, ch. 08, 391–421.
16. Halbe, D. A Qualitative Analysis of Differences in the Features of Telephone and Face-to-Face Conferences. In Association for Business Communication (2008).
17. Harrison, S., Tatar, D., and Sengers, P. The Three Paradigms of HCI. In alt.chi (2007).
18. Hellriegel, D., and Slocum Jr., J. Organizational Behavior, tenth ed. South-Western College Pub, 1995.
19. Hinds, P., and Bailey, D. Out of sight, out of sync: Understanding conflict in distributed teams. Organization science (2003), 615–632.
20. Hoffman, L. R. Applying Experimental Research on Group Problem Solving to Organizations. The Journal of Applied Behavioral Science 15, 3 (July 1979), 375–391.
21. Jayagopi, D. B. Computational modeling of face-to-face social interaction using nonverbal behavioral cues. PhD thesis, Ecole Polytechnique F´ed´erale de Lausanne, 2011.
22. Kim, T., Chang, A., Holland, L., and Pentland, A. S. Meeting mediator. ACM Press, New York, New York, USA, Nov. 2008.
23. Maes, P. How to Do the Right Thing. Tech. rep., Cambridge, MA, USA, 1989.
24. Olson, G. M., and Olson, J. S. Distance matters. Hum.-Comput. Interact. 15, 2 (Sept. 2000), 139–178.
25. Potter, R. E., Balthazard, P. A., and Cooke, R. A. Virtual team interaction: assessment, consequences, and management. Team Performance Management 6, 7/8 (2000), 131–137.

26. Rienks, R., Nijholt, A., and Barthelmess, P. Pro-active meeting assistants: attention please! AI Soc. 23, 2 (Aug. 2008), 213–231.
27. Schegloff, E. A., and Turner, J. H. Accounts of Conduct in Interaction. In Handbook of Sociological Theory, J. H. Turner, Ed., Handbooks of Sociology and Social Research. Springer US, 2001, ch. 15, 287–321.
28. Selker, T. COACH: a teaching agent that learns. Communications of the ACM 37, 7 (July 1994), 92–99.
29. Selker, T. Understanding considerate systems UCS (pronounced: You see us). In 2010 International Symposium on Collaborative Technologies and Systems, IEEE (2010), 1–12.
30. Smith, E. E., and Kight, S. S. Effects of feedback on insight and problem solving efficiency in training groups. Journal of Applied Psychology 43 (1959), 209–11.
31. Vinciarelli, A., Pantic, M., Bourlard, H., and Pentland, A. Social signal processing: state-of-the-art and future perspectives of an emerging domain. In Proceeding of the 16th ACM international conference on Multimedia, MM '08, ACM (New York, NY, USA, 2008), 1061–1070.
32. Weber, M. Economy and Society: An Outline of Interpretive Sociology. University of California Press, USA, 1978.
33. Yankelovich, N., Walker, W., Roberts, P., Wessler, M., Kaplan, J., and Provino, J. Meeting central: making distributed meetings more effective. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, CSCW '04, ACM (New York, NY, USA, 2004), 419–428. 34. Young, S. Cognitive User Interfaces. IEEE Signal Processing Magazine 27, 3 (May 2010), 128–140. 10