Designing and Evaluating Meeting Assistants, Keeping Humans in Mind

Patrick Ehlen
Raquel Fernández
Matthew Frampton

Center for the Study of Language and Information
Stanford University, 210 Panama Street
Stanford, California, USA
{ehlen, raquel.fernandez, frampton}@stanford.edu

Abstract. Meeting assistants pose some interesting and unique challenges to the enterprise of software design and evaluation. As the technology reaches greater levels of development, we must begin to consider methods of evaluation that reach beyond regarding meeting browsers as signal replay and information search tools, and begin to assess the dimensions in which meeting assistants and browsers can augment or hinder human cognition and interaction. Some of these dimensions are considered, inasmuch as they were encountered during development of the DARPA CALO Meeting Assistant and Meeting Browser.

Keywords: meeting browser, meeting assistant, multimodal, evaluation, design, user requirements, CALO.

1 Introduction

Meetings are an important aspect of modern life. Sometimes people miss a meeting or forget exactly what happened in one, and it would be handy if those people could just ask a computer to tell them the things they need to know. It would be even handier if that computer could find and re-create the relevant parts of the meeting, from any perspective in the room, like computers do on *Star Trek*. Unless, of course, the computer goes berserk and starts to make up things that never happened. But let's not worry about that yet.

Of more pressing concern is the question of how we can develop meeting assistance tools that render meetings more productive and the information exchanged in them more durable and accessible. From a high-level perspective, there are two dimensions on which meeting participants may be aided: *participation* and *memory*. The dimension of participation includes finding ways to help people to interact efficiently and constructively and to exchange the right kinds of information at the right moments. The dimension of memory includes finding ways to make meeting information "stick," so to speak, either by making it more accessible in people's heads (their "organic memory") or by making it more accessible somewhere else, such as in the notes they've taken (or "prosthetic memory") [1].

The relation between these two dimensions of participation and memory is fairly orthogonal—which means they don't always work hand-in-hand, and a tool that helps one does not necessarily help the other. Efforts at aiding participation can hinder memory by failing to encourage information consolidation. Likewise, tools designed to help memory can hinder participation. But participation and memory can sometimes be tapped in tandem, as happens when videoconferencing tools aid memory by promoting cross-modal encoding. So when faced with the task of evaluating meeting assistance tools, we cannot in good conscience invoke only one of these dimensions as grounds for appraisal. Rather, as designers of meeting assistants and meeting browsers, we must consider how both of these dimensions of participation and memory can be evaluated, and develop tools and methods for striking the right balance between the two, given the varying circumstances in which such tools may be used.

One further consideration should not be missed when designing tools that broker in human interaction and language: Any tool that interacts with people ultimately has the potential to change the way those people behave, and thus may alter the effectiveness of—or even break—the tool itself. For example, a system that identifies people's spoken commitments during a meeting and creates a record of them may eventually cause people to be more specific and deliberate when speaking about commitments, or may make them less likely to commit to things using speech. In the same way that laptops, PDAs, and presentation software have changed the way people act during meetings over the past few decades, so will the meeting assistant technologies we develop today change the behavior of tomorrow's meeting-goers. Any technology that aims to endure must be flexible enough to adapt to changing patterns of interactive behavior.

The aforementioned sensitivity of such tools to the vagaries and reactivity of human behavior throws a spanner in the works of typical software development cycles that tend to progress iteratively, basing the next iteration's set of development requirements on the failures of the prior iteration. This is the *iterative Catch-22* of development for meeting assistants, and the only way out of the mire is to design tools that adapt to their circumstances the way people do. Let us keep these thoughts in mind as we review a couple instances of interfaces designed for the DARPA CALO Meeting Assistant (CALO-MA).

2 Meeting Assistance Tools

Meeting assistance tools come in two flavors: *online* and *offline*. An online tool allows participants to interact with it during the meeting. This would include anything from traditional notepads and whiteboards to a virtual secretary that interacts with the participants. An offline tool, by contrast, is designed to be used at some point outside the meeting. It might help participants prepare for a meeting, or it might allow them to revisit aspects of the meeting after it's finished. Such tools would include browsable video recordings or transcripts, or a daemon that quietly identifies the tasks people agree to do during a meeting and places them on participants' to-do lists when they return to their desks. (A review of online and offline approaches can be found in [2]).

Each of these two flavors of meeting assistance tools has its advantages and disadvantages. Online tools have the advantage of allowing people to specify and lock in information while it's fresh in their minds, and can foster immediate feedback regarding the quality and accuracy of information as it is stored. But the presence of the technology in the ongoing meeting can distract from normal interaction and from the decision-making process. As a simple example, when people take notes during a meeting, they must either tune out of a conversation for a moment, or halt the conversation process as they write their notes. Offline interfaces, on the other hand, have the advantage of allowing meeting participants to focus on participation, and encourage feedback to happen at a more leisurely pace (such as later in the day, when participants return to their desks). If an offline tool is part of a wider suite of applications, it can help to integrate information established during meetings with other desktop tools, such as to-do lists, calendars, e-mail, or project planners. However, many management-level workers of our era rightfully smirk at the idea of attending to meeting-related interfaces after "returning to the desk," since their workdays often consist of a series of one meeting after another, with no desk to be seen until the end of the day, by which time a great deal of information may be degraded or completely forgotten.

Each of these options poses challenges to design and evaluation. We'll first consider the offline interface experience for CALO-MA.

2.1 The CALO Offline Meeting Assistant and Browser

As part of a wider DARPA CALO research project effort, the CALO-MA group inherited a mandate to design a meeting assistance system that would not only be effective and usable, but also would learn to improve over time, preferably in a personalized manner. This mandate nudged development toward two simultaneous efforts: (1) an effort to create models of speech and behavior that begin as functioning generalized models, but can adaptively evolve into personalized ones; and (2), an effort to solicit and incorporate user feedback that can retrain those models and improve and personalize them over time. These models include ASR language models, gesture and handwriting models, topic models, and models to classify sets of ASR-transcribed spoken utterances into dialogue acts, question and answer pairs, action items, and decision discussions.

A second, self-imposed mandate of CALO-MA was that the system under development should not include any type of in-meeting dialogue system, since such a system could prove disruptive to the natural flow of meeting dialogue. So maintaining natural *participation* was prioritized over the possible benefits of having a system that participants could explicitly address. Each participant is given a wireless headset that sends audio to a VoIP client. The VoIP client also provides a small suite of software collaboration tools, such as chat, notes, and a shared whiteboard. The system is designed to work equally well for remote, distributed meetings as for meetings carried out with all participants at the same table.

Post-Meeting Process. When participants finish their meeting, audio is delivered to a server that begins a chain of processes, such as producing an ASR transcript and

detecting topics, question-answer pairs, and action items. When complete, an e-mail is sent to the participants, who may then review the transcript and extracted information in an offline meeting browser. This browser displays the transcript, with audio playable from any point. But more importantly, it displays hypotheses for the distilled information that users would be likely to want to retain a record of, such as action items and decisions (see [4] and [5] for more extensive descriptions of the CALO-MA meeting browser and its workings).

These hypotheses, as results of machine learning, are far from perfect. So the meeting browser is designed to harvest the implications of ordinary user actions as *implicit user feedback* that can be used to retrain classifier models without explicitly asking the user for feedback. For example, action items from the browser that a user adds to a to-do list are marked as valid positive instances for future retraining, while action items that are explicitly rejected are tagged as negative instances for retraining. If a user changes the description or responsible party of an action item, these actions are also harvested for future retraining, so the system will improve over time. Action item detection models retrained on even a few meetings' worth of feedback data can show reasonable improvements [5].

Humans in the Loop. Even though this offline system is designed to be unobtrusive and essentially invisible during the meeting process, aspects of the system's design had a discernable effect on people's behaviors during meetings, which in turn affected the system's behavior—resulting in the aforementioned iterative Catch-22. For example, an action item detection system was initially trained on transcripts from a diverse set of meetings (collected from the ICSI and ISL corpora, as well as some meetings recorded at SRI and CSLI). Participants' action items were detected by this model during test meetings at SRI and posted to an "Action Items" section of the offline meeting browser which participants could review a few hours after each meeting.

But the action item detection system did not work as well as expected for some participants, who expressed surprise after they diligently and explicitly stated declarations of action items during meetings, using statements along the lines of, "So here is an action item for you, to write up a plan before the next meeting." Not surprisingly, the original utterance data used to train the action item classifiers did not tend to contain such explicit statements of task commitments; and the words "action item" were not present in a single training meeting. So why did these new participants suddenly speak this way? Most likely because they were now aware that action items were being explicitly noted by some external entity, and because the meeting browser itself displayed a rather prominent section labeled "Action Items," which primed the participants to use that term. The detection system thus needed to be retrained on a set that included meetings that contained these types of utterances, so such explicit talk about "action items" would also be detected as such.

By contrast, a later iteration of the system actually worked better than expected, but for a similar reason: For each action item detected, the meeting browser displayed fields for the person(s) responsible for the action item, as well as the timeframe in which the action item should be completed, and these fields were populated when such information could be identified. But the presence of those fields in the browser prompted meeting participants to produce more utterances that specified not only

what tasks needed to be done, but who would do them and when. Since these types of utterances contribute to the success of overall action item detection, this unexpected change in behavior led to better detection of action items than prior iterations [5].

2.2 Online Meeting Assistants

If people's behavior during meetings can be influenced by interfaces that do not actively participate in the meeting, we must wonder about the extent to which their behavior will change when different types of meeting assistance interfaces are introduced into the meeting room itself, and what effects these interfaces will have on participation and memory. As mentioned earlier, the tools people currently use during meetings—such as laptops and notepads—incur a certain level of cognitive load which can require people to "check out" of the meeting (even if only briefly) as they attend to the tools that promise to help them remember more information down the line.

The current state of technology, as demonstrated by the existing CALO-MA system, shows potential for different types of in-meeting interfaces that could help or hinder both participation and memory. For example, consider the cognitive load incurred by a person engaged in ordinary note-taking: While listening, that person must select information from dialogue that is salient, then distill and consolidate that information into a sensible chunk, and must finally exert the language and motor skills required for production of that distilled information onto a piece of paper (or keyboard). It's no wonder that many people have a hard time participating in a conversation while simultaneously taking notes. But if a real-time ASR transcript were generated during a meeting in progress and scrolled before each participant, participants could take notes simply by marking or highlighting the portions of the transcript they wish to revisit later. Such a process could aid participation by removing the cognitive load involved in note-taking; only listening and selection would be required. (Note that lawyers and judges in the courtroom have had access to this sort of technology advantage for years, thanks to digital networks that link their stations to electronic transcripts produced by professional stenographers.)

An even simpler interface might be to give each participant some type of "button" which could be pressed whenever a salient event happens during the meeting. Once a region of the meeting is indicated as containing salient information, machine learning techniques could attempt to extract that salient information and save it for the participant to access later, or even to act on it in some way. In either case, participants are freed from a good-deal of "record-keeping" and allowed to engage in more productive interactions.

But would these interactions necessarily be more productive? From the standpoint of participation, such interfaces may indeed allow people to participate more. But more is not always better. From the standpoint of the dimension of memory, we may arrive at a different perspective: Because such interfaces can provide a substitute for the cognitive process of consolidation that would normally take place during note-taking, they could actually lead to meetings where people talk more, but walk away remembering less.

Of course, this type of question can only be answered through an empirical study, and an experiment designed to provide that answer is now underway. Cognitive measures for participation and cognitive load can be obtained through both subjective measures, such as questionnaires given to meeting participants, and objective measures, such as comparative statistics on the contributions people make during meetings when using different types of interfaces. For measures of memory, the best method may be to test how well people remember the things that happened during a meeting by asking them to recall events and decisions at later intervals.

3 Final Thoughts

This brief discussion has covered some real-world attempts to develop meeting assistant and browser interfaces over the past two years, as part of the CALO-MA project. We have discussed possibilities for both offline and online interfaces, and looked at how the dimensions of participation and memory must ultimately figure into evaluations of such interfaces, pointing the way to a (perhaps foggy) realm of evaluation beyond gold-standard annotations and F-scores. We have also discussed some examples of how the typical software design process can result in an iterative Catch-22, which hints at a need for design methods that treat meeting assistant software as part of the interactive process, and not an appendix to it. Only software that can adapt to the behaviors and variations of its users will prove flexible enough to avoid that iterative loop.

Other methods of evaluation for meeting browsers have been put forward, such as the BET [6,7], and these methods work well for evaluating browsers of automatically-generated meeting information repositories that will be searched by users who did not necessarily participate in the meeting. But when it comes to evaluating tools that are more “embedded” in the process of meeting participation and incorporating information and decisions into the everyday work cycle, there are many more possibilities left to consider.

Acknowledgments. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the Department of Interior-National Business Center.

References

1. Kalnikaite, V., Whittaker, S.: Software or Wetware? Discovering When and Why People Use Digital Prosthetic Memory. In Proc. CHI07, pp. 71-80. ACM Press (2007).
2. Rienks, R., Nijholt, A., Barthelmess, P.: Pro-Active Meeting Assistants: Attention Please! AI & Soc. 2008.
3. Voss, L. L., Ehlen, P. and The CALO Meeting Assistant Team: Multimodal Meeting Capture and Understanding with the CALO Meeting Assistant. In Proc. MLMI (2007).

4. Ehlen, P., Purver, M., & Niekrasz, M. A Meeting Browser that Learns. In Proc. AAAI (2007).
5. Ehlen, P., Purver, M., Niekrasz, J., Lee, K., & Peters, S.: Meeting Adjourned: Off-line Learning Interfaces for Automatic Meeting Understanding. In Proc. IUI (2008).
6. Wellner, P., Flynn, M., Tucker, S., Whittaker, S.: A Meeting Browser Evaluation Test. In Proc. CHI05, ACM Press (2005)
7. Tucker, S., Whittaker, S.: Accessing Multimodal Meeting Data: Systems, Problems and Possibilities. In: Bengio, S., Bourlard, H. (eds.) MLMI 2004. LNCS, vol. 3361, pp. 1-11. Springer, Heidelberg (2005)