Flexi-modal and Multi-Machine User Interfaces

Brad Myers, Robert Malkin, Michael Bett, Alex Waibel, Ben Bostwick,
Robert C. Miller, Jie Yang, Matthias Denecke, Edgar Seemann, Jie Zhu,
Choon Hong Peck, Dave Kong, Jeffrey Nichols, Bill Scherlis
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bam@cs.cmu.edu 1-412-268-5150
http://www.cs.cmu.edu/~cpof

Abstract

We describe our system which facilitates collaboration using multiple modalities, including speech, handwriting, gestures, gaze tracking, direct manipulation, large projected touch-sensitive displays, laser pointer tracking, regular monitors with a mouse and keyboard, and wirelessly-networked handhelds. Our system allows multiple, geographically dispersed participants to simultaneously and flexibly mix different modalities using the right interface at the right time on one or more machines. This paper discusses each of the modalities provided, how they were integrated in the system architecture, and how the user interface enabled one or more people to flexibly use one or more devices.

Keywords: Multi-modal interfaces, speech recognition, gesture recognition, handwriting recognition, gaze tracking, handhelds, personal digital assistants (PDAs), laser pointers, computer supported collaborative work (CSCW)

1. Introduction

We are working to provide an effective collaborative, integrated environment for situational awareness. The target environment is a military Command Post of the Future (CPoF), where the commander and staff work. The environment must be flexible and support multiple modalities to serve the needs of the commander. We envision that the commander stands or sits in front of large displays showing maps and other status information, and that speech, pointing, gestures and other natural techniques are used to control the views and to provide input to the computers. A number of support staff are also involved, and information and control is fluidly shared. Handhelds and other devices are used for private work, which is easily shared with the group when appropriate. We created a prototype of what a system for this command post might be like, shown in Figure 1.

Some important requirements for the command post of the future follow. Many of these requirements are also relevant to multi-modal systems for businesses and homes.

Figure 1: One of the authors working on Battleboard projected onto touch-sensitive SMARTBoard. Visible are the map view (left), a table and chart (center), and the iconic recognizer (bottom right).

- **Flexible modality choice:** Users need to be able to speak, gesture, tap on a SMARTBoard interactive whiteboard, use a mouse, point with a laser pointer, type on the keyboard, and use handwriting recognition, in whatever combination they desire. They also need to be able to rapidly and fluidly change modalities, even during a single interaction. This is required because the command post can be a noisy and chaotic place, and users may start an interaction by speaking, but have to switch to handwriting because a loud airplane passes overhead. We call this property "flexi-modal."
- **Multiple-machine user interfaces:** Users may be sitting at workstations, standing up at large displays, or walking from place to place, and still need to interact in all situations. Users will be carrying various technologies, such as mobile phones and personal digital assistants (PDAs) such as Palms or PocketPCs. Interactions should take advantage of these technologies, and should be adapted to the capabilities of the devices available. In particular, when users are standing or walking, they should be able to use their handheld PDAs to augment their interaction with the larger displays.

- **Support for multiple people:** Most tasks in the command post are collaborative, and the system should facilitate the exchange of information and control. People take turns talking, gesturing and interacting with the information, and the system should provide mechanisms that enhance rather than inhibit fluid interaction among the people involved.

The work reported in this paper addresses all of these requirements. It is novel on a number of levels. We believe that it is the first collaborative integrated system to successfully and seamlessly combine speech, gestural commands, handwriting, gaze recognition, laser tracking, direct manipulation, and multiple handheld devices. Furthermore, we have developed a novel technique for multiple-machine user interfaces, where handhelds can be used to get more information about the data projected on the wall, which we call *private drill-down of public information*. Finally, our inclusion of both command-and-control and large-vocabulary speech recognition is novel in that it allows us both to control the information display via speech and to capture and record human-human and human-machine interactions for later perusal and summarization.

2. Related Work

Related work in multi-modal systems dates back to the original "Put-That-There" system [2]. Recent work includes the Rasa system [9], which is another multi-modal interface for the Command Post of the Future. Rasa uses a commercial speech recognition system and supports gestural interfaces on a handheld device. Their studies show that using a multi-modal map interface is more efficient than a conventional direct manipulation interface [3] [9].

eClass [1] allows instructors to write on a blank electronic whiteboard or on top of prepared slides. The electronic annotations, audio, video, and even Web browser activity, are all automatically recorded and time-stamped. By capturing these events, the system can later recreate the lecture experience. No speech or gestures are supported, however.

Our gaze tracking system is related to the "look-to-talk" system from MIT, which was shown to be much preferred over "push-to-talk" in a user study [13].

The integration of handhelds and laser pointing with PC-based applications has been studied by various groups. For example, the Stanford iRoom project is investigating a large wall display with handhelds and a laser pointer, using special gestures and pie menus for the interaction [18]. Rekimoto has studied how to move information fluidly using a "pick-and-drop" metaphor between handhelds and a whiteboard [14] and he introduced "hyperdragging" as a way to move data by dragging it from one device to another [15]. The issue of public and private displays has been explored in the GroupLab system [7], but it was restricted to a single modality.

However, none of these groups have tackled such an ambitious project by combining all of these modalities into a single architecture that also supports collaboration using multiple machines simultaneously.

3. Command Post Task

The command post task that we addressed in this work is a situation / information awareness task, in contrast to a planning task. As such, our activities focus on providing natural ways for users to extract information contained in a visual display. This visual display, along with databases and models for interacting with the display, was provided by a local company, MayaViz, Inc. This application, which we call the *battleboard*, is written in Java. It provides maps showing units moving over time, along with tables and charts of the units' attributes (see Figure 1). It initially provided only a standard desktop GUI interface, so our work for the CPoF task involves supplying multiple, flexible modes of interaction with the MayaViz battleboard using their supplied programming interface.

4. Speech, Handwriting and Gestures

In order to provide full flexi-modal support for the command post, we built several independent input services targeted for military users. These include command-and-control speech recognition, iconic gesture recognition along with support for integrated selection gestures, handwriting recognition, typed input, and passive entity tracking.

- **Command and Control Speech Recognition.** For direct speech manipulation of the display, we provide a command-and-control system based on the XCalibur speech recognition engine [5]. The engine is a speaker-independent, fully-continuous Hidden Markov Model (HMM) speech recognizer which implements the Java Speech API (JSAPI). Using the Java Speech Grammar Format (JSGF), we built a small (about 1000 word) command-and-control recognizer specifically for the CPoF task. The grammar covers several types of interaction: object selection by unit type, unit name, unit affiliation, or a combination of these attributes; querying of unit attributes such as morale, firing range, or mobility; information transfer between visualizations (maps, tables, and charts); and manipulation of the visualization settings. We support the use of context and anaphoric unit reference ("these units") by assuming that such anaphora always refers to the units currently selected.

- **Integrated Gestures**. Using methods provided by the battleboard interface, we provide integrated selection gestures on both the SMARTBoard interactive white-

board and the handhelds. For example, circling units or tapping on units causes them to be selected.

- **Handwriting Recognition**. The visualization framework provided by MayaViz includes the ability to attach annotations to specific units at specific times. Users can place these annotations in such a way that they could communicate important features or messages to other users of the CPoF in an asynchronous fashion. We enhanced the annotation procedure by providing a handwriting interface, which is more convenient and natural than typing, especially for users on the SMARTBoard. We used the NPen system [8], a large-vocabulary, writer-independent, connected-word, neural-net handwriting recognizer as this interface. In order to incorporate handwriting and integrated focus gestures into the same system, we inserted a mode switch into the battleboard, though in principle it is possible to add a preprocessing component to the handwriting recognizer to classify input ink events as annotations or gestures.
- **Iconic Gesture Recognition**. As an alternative to explicit focus gestures, or spoken focus statements like "Select the enemy artillery units," we provide a significantly faster and easier iconic gesture recognition for unit selection. We constructed from the ground up an extensible template-based icon recognition system which allows users to draw a unit type in a separate input window (shown at the bottom right of Figure 1), have it recognized, and have all units matching that type be selected. The icon recognition system recognizes enemy and friendly unit types of several main categories (armor, reconnaissance, anti-aircraft, infantry) and several modifiers (mechanized, wheeled, airborne), yielding a large set of recognizable unit types. The gestures can be drawn directly onto the SMARTBoard, with a laser pointer, or with a remote handheld with a stylus (see sections 7 and 8). In a small user test using untrained individuals on a data set of 41 gestures, our gesture recognizer was 83.48% accurate. Accuracy improves significantly with training as individuals learn how to draw the gestures.
- **Passive Entity Tracking**. In addition to direct command speech, users often wished to discuss the contents of the display without interacting with the CPoF system. Since these human-human interactions may contain valuable information about the scenario, we capture them for transcription by a conventional large-vocabulary conversational speech recognition (LVCSR) system. This system was based on the Janus Speech Recognition Toolkit (JRTk) [6], and was modified for vocabulary coverage. Speech from users is continually tracked and could later be inspected by use of the Meeting Browser system [16, 17]. With the Meeting Browser, users of the CPoF can view transcripts, summaries, and topic shifts of CPoF interactions, as well as static visual snapshots of the battleboard state over time. In addition to this after-action review usage pattern, we also provide the ability to passively track unit types as they were being discussed. In this way, when users of the CPoF mentioned units by type or name, the CPoF display would continually be updated by moving those units into focus.
- **Typed Input**. In addition to command speech, we also provide the ability to type commands to the display for those times when speech recognition is unavailable or impractical.
- **Fleximodality.** Rather than force the user into one modality per interaction, we provide multiple choices for each type of interaction. Manipulation of the display can use speech, typed input, or direct manipulation provided by the display itself. Annotations can be made by typing or handwriting. Unit selection can be done by speech, direct gesture, or iconic gesture. Since we wanted user interactions with the battleboard to be as natural as possible, this freedom of choice is essential.

We found that the original direct manipulation interface was adequate for most tasks, but often required long sequences of actions to obtain the desired results. For example, in order to view an attribute of a set of units in a table required multiple actions: creating a table, dragging the desired units to the table, and then adding that attribute to the table. Worse, if we knew which units we wanted to appear in the table, but did not know where they were on the map, we had to find them first. This is not a major obstacle if the units whose attributes we want to query are located in roughly the same area. However, if we want all instances of a specific type of unit (e.g. air defense), or a specific unit by name (e.g. 1st armor), the process becomes time-consuming and error-prone. Furthermore, it may require use of the keyboard. We wanted this type of action to be achievable in one or two steps and without use of the keyboard. For most interactions this is now the case. For the table display example, we can use speech only for some cases (e.g. "Place the artillery units in a table; Show combat power in the table."), and speech combined with a focus gesture for others (e.g. a direct focus gesture followed by "Place these units in a table; Show combat power in the table."). We found this kind of interaction to be superior to the direct manipulation method, and further, superior to speech alone.

5. Gaze Tracking

In order to interact naturally with the battleboard, we wish to operate the speech recognition system in always-on mode rather than push-to-talk mode. The XCalibur engine [6] provided methods for setting silence thresholds in order to avoid transcribing noise; however, these

thresholds prove inadequate in that we often wish to speak without speaking to the battleboard system. This would force us to use a push-to-talk style of interaction. We were able to remove the push-to-talk requirement by using a camera to track the main user's gaze. In this way, we provided a mode switch that was "on" when the user was looking at the battleboard, and "off" otherwise. This mode switch was used to turn the speech recognition system on or off, though it could also be used to switch from command speech to passive speech.

The tracking system was implemented with a neural net trained to recognize head pose and determine whether the pose meant "user looking at battleboard" or "user looking elsewhere."

6. Dialog

For the CPoF task, we used a shallow dialog system based on the parse returned by the speech recognizer. We use the tag facility of the Java Speech Grammar Format (JSGF) to encode meta-level information about the utterances. These meta-level tags are then used to determine what action the user is requesting and what objects are to be affected.

7. Laser Tracking

Speech interfaces enable *interacting at a distance*, but this is not possible with handwriting, gesture or direct manipulation using a mouse or touch screen. A common way to point to objects that are on the screen during presentations is using a laser pointer, so it is natural to think about how computer tracking of a laser pointer could be used in the command post to enable interacting at a distance. We performed various studies of laser tracking, and measured a number of important properties [11]. For example, the unsteadiness of people's hands means that targets must be about 10 pixels across, and interactions normally must be fairly slow. In fact, we found that tapping on the SMARTBoard was about twice as fast as using the laser pointer for a selection task [11].

However, once integrated into the command post system, we did find that the laser tracker is useful in some situations. People anywhere in the room can move the cursor to the items of interest, and can even perform some simple gestures and other interactions. However, detailed interactions, such as selecting from menus or drawing icons, are difficult to perform using the laser pointer from across the room.

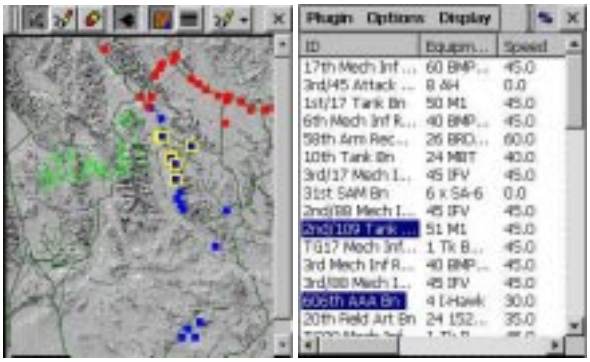
8. Handhelds

Staff people are constantly walking into, out of, and around the command post, and often carrying some kind of handheld device, such as a portable phone or PDA. We are investigating a number of ways that using the handheld can augment the other modalities.

One concept we are exploring is *private drill-down of public information*. When multiple people are using a single large display, it may not be appropriate for one person to usurp it for private work. Therefore, we provide a unique and fluid way for a user to get the appropriate content from the shared display to a private handheld display. Figure 2(a) shows the handheld representation of a map with some units highlighted and a hand-drawn annotation. Figure 2(b) shows the "drill-down" table displaying information about particular units. When private work is complete and should be shared with the group, then the displays can be resynchronized. Multiple people can be separately drilling down to different information on their private handhelds further facilitating collaboration.

An important issue with private and public displays is the level of coordination to apply between them. Should selections (highlights) be synchronized so all displays show the same units selected? What about annotations (drawings)? When units move, are deleted, or otherwise change properties, should that be reflected on all displays? When a display is scrolled or zoomed to show certain items, or filtered to temporarily hide some items, should the other displays also be focused on the same view?

We decided that providing individual control over all of these options would be too confusing, and so we just provide two modes for each display. When *connected*, selections, annotations, and all data changes are kept consistent between the public and handheld displays, but scrolling and zooming is always independent for each display. When *disconnected*, the user can freely change the handheld's views, selections and annotations, which are private. However, data changes to unit properties and positions are still reflected, so it really is not really disconnected from the data source. When the user switches from disconnected to connected, the system checks for differences between the handheld's view and the shared view. We decided that providing individual control over each change would be too complex, and so we just allow the user to choose between having the public view override the handheld, or the handheld override the public



(a) (b)

Figure 2: Handheld displays of (a) the map and (b) a table of data. These appear on a Compaq iPaq's screen.

view. Future experience and testing will be required to refine these user interface design decisions.

In addition to the custom-written CPoF application for the PC and handheld, we also are investigating some techniques for sharing information and control between handhelds and public displays even when running commercial off-the-shelf (COTS) applications on the PC. *Semantic snarfing* [12] is a technique for copying ("snarfing") the information from the shared screen onto the handheld screen in a way that preserves the meaning ("semantics") on the handheld. For example, text on the shared screen is reformatted as readable and editable text on the handheld. Semantic snarfing is very useful with laser pointing, because the laser pointer is good for showing the general area of interest, which can then be snarfed onto the handheld, and then detailed work can be performed in an efficient manner on the handheld.

9. Architecture and Integration

Integration of multi-modal input with the MayaViz battleboard is handled through several transaction managers (as shown in Figure 3):

- **Focus Gestures.** Focus gestures are implemented by the battleboard, with no extra interpretation.
- **Command Speech, Passive Speech, Iconic Gesture, Handwriting Recognition, and Typing.** These input modalities are handled by a single transaction architecture. In this architecture, each input service is required to implement a simple server interface (labeled "Multimodal Server" in Figure 3) which sends text over a socket to the Multimodal Interpreter client. This client then parses the input command and invokes the appropriate method in an executive object (labeled "Executor" in Figure 3) which makes direct calls to the MayaViz battleboard API. For command speech and typed input, the text is passed directly. For iconic gesture and passive selection, we first translate the indicated units into a focus command (i.e. the result "enemy artillery" from the icon recognition system is translated to "focus on the enemy artillery units."), and then send that text via a socket to the client. For handwriting annotation, the recognized text is converted into an annotation command and sent to the client. All modalities except handwriting have their input streams implemented independently of the battleboard. For handwriting, the battleboard provides a "handwriting" mode specifically designed to pass the captured ink trajectories via a socket to the NPen system.

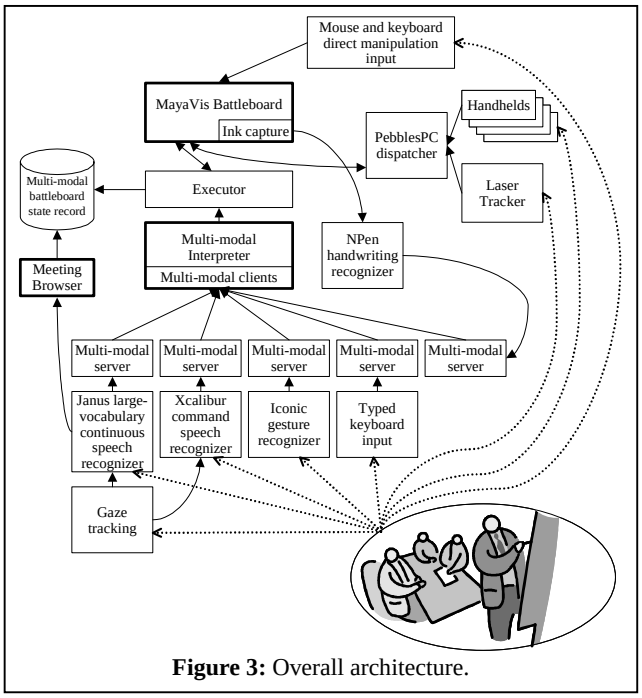


Figure 3: Overall architecture.

- **Meeting Browser:** As the speech is recognized by Janus, it is also stored for later viewing and summarization by the Meeting Browser. Also recorded are commands and screen shots from the battleboard.
- **Handhelds and Laser Tracking:** The PebblesPC dispatcher [10] accepts input and distributes the output to one or more handhelds. It connects to the same battleboard API as the other recognizers. The Laser tracker also uses the PebblesPC dispatcher, which converts the detected laser positions so that they appear to the battleboard to be regular mouse positioning events.
- **Multi-Machine and Multi-Site Capabilities:** The integration architecture was designed to pass information via sockets so that computational load could be spread over multiple machines. In particular, we wished to avoid running the three most computationally intensive modules – the battleboard, the command-and-control speech recognizer, and the large-vocabulary speech recognizer – on the same processor. As a result, the only components that must be co-located are the battleboard, executor, and interpreter modules. The other components can be run from any location. Another advantage of this architecture is that it allows for collaboration between non-co-located sites. So long as their local input devices implement the Multimodal Server interface and know the socket address of the Multimodal Interpreter, any user can interact with the battleboard using multiple modalities.

10. Observations

In informal evaluations, the flexi-modal command-and-control interface worked well. The ability to choose the favored modality for the task improved system utility over direct manipulation or speech alone. For example, item selection could be done by focus gesture (circling), speech, or iconic gesture. Focus gestures were most appropriate when the user was seeking to highlight an area.

Speech commands were most appropriate when looking for a specific unit by name (e.g. "117th infantry"). Finally, iconic gestures were most appropriate when seeking specific unit types.

We initially had concerns about synchronization of gesture and speech. However, since all gestures were in essence focus gestures, no problems occurred. Speech commands involving attribute queries were assumed to apply to whatever objects were currently in focus, or contained focus commands themselves. That is, "Give me the combat power for these units" would rely on whatever units were in focus, while "Give me the combat power for the enemy infantry units" would modify system focus before addressing the attribute query. Further, when focus shifted, attribute display shifted as well. Thus, if an attribute query were addressed before a focus shift request, the system would end up in the desired state.

11. Future Work and Conclusions

Although our funding for the CPoF project has ended, there are many interesting issues that could be investigated in the future. The coordination of multiple people using the shared public display and various private, handheld displays, along with a shared audio space (for speaking) brings up many interesting research questions. The accuracy of all of the recognizers will be improving separately, but it is also useful to investigate the benefits of using the modalities together, for example to improve overall success rates.

Our experiences with the CPoF project lead us to believe that fleximodal, distributed, multi-user interfaces can be used to enhance any application that involves human exploration, presentation, and sharing of data. A good example of such an application is a distributed presentation tool in which a standard PowerPoint presentation is augmented with datasets and tools. Together with methods for rapid design of dialogue systems [4], we feel that the tools and architecture described here can easily be used for such applications. Large-scale projects like the CPoF provide useful information on the issues that will arise in the integrations that will be needed for real-world use of multi-modal systems as they become generally practical.

Acknowledgements

This research was performed in part in connection with contract number DAAD17-99-C-0061 with the U.S. Army Research Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. We are also grateful for equipment donations by Symbol Technologies, Lucent, and SMART Technologies, Inc.

References

[1] Abowd, G., "Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment." *IBM Systems Journal*, 1999. **38**(4): pp. 508-530.

[2] Bolt, R.A., ""Put-That- There": Voice and gesture at the graphics interface." *Computer Graphics*, 1980. **14**(3): pp. 262-270.

[3] Cohen, P.R., *et al.* "QuickSet: Multimodal Interaction for Distributed Applications," in *Proceedings of the 5th ACM International Conf. on Multimedia*. 1997. Seattle: pp. 31-40.

[4] Denecke, M. "Object-Oriented Techniques in Grammar and Ontology Specification," in *The Workshop on Multilingual Speech Communication*. 2000. Kyoto, Japan: pp. 59-64.

[5] Finke, M., *et al.* "Modeling and Efficient Decoding of Large Vocabulary Conversational Speech," in *Proceedings, Eurospeech-99*. 1999. Budapest: pp. 467-470.

[6] Finke, M., *et al.* "The JanusRTk Switchboard/Callhome 1997 Evaluation System," in *The DARPA Large Vocabulary Conversational Speech Recognition Hub5e Workshop*. 1997. Baltimore

[7] Greenberg, S., Boyle, M., and Laberg, J., "PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal." *Personal Technologies*, 1999. **3**(1): pp. 54-64. March.

[8] Jaeger, S. "NPen++: An On-line Handwriting Recognition System," in *7th International Workshop on Frontiers in Handwriting Recognition*. 2000. Amsterdam: pp. 249-260.

[9] McGee, D.R., *et al.* "Comparing Paper and Tangible, Multimodal Tools," in *ACM CHI'2002 Conference Proceedings: Human Factors in Computing Systems*. 2002. Minn, MN: pp. 407-414.

[10] Myers, B.A., "Using Hand-Held Devices and PCs Together." *Comm. of the ACM*, 2001. **44**(11): pp. 34-41.

[11] Myers, B.A., *et al.* "Interacting At a Distance: Measuring the Performance of Laser Pointers and Other Devices," in *ACM CHI'2002 Conference Proceedings: Human Factors in Computing Systems*. 2002. Minn, MN: pp. 33-40.

[12] Myers, B.A., *et al.* "Interacting At a Distance Using Semantic Snarfing," in *ACM UbiComp'2001*. 2001. Atlanta, Georgia: pp. 305-314.

[13] Oh, A., *et al.* "Evaluating Look-to-Talk: A Gaze-Aware Interface in a Collaborative Environment," in *Extended Abstracts for CHI'2002: Human Factors in Computing Systems*. 2002. Minneapolis, MN: pp. 650-651.

[14] Rekimoto, J. "A Multiple Device Approach for Supporting Whiteboard-based Interactions," in *SIGCHI'98: Human Factors in Computing Systems*. 1998. Los Angeles, CA: pp. 344-351.

[15] Rekimoto, J. and Saitoh, M. "Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments," in *SIGCHI'99: Human Factors in Computing Systems*. 1999. Pittsburgh, PA: pp. 378-385.

[16] Waibel, A., *et al.* "Advances in Automatic Meeting Record Creation and Access," in *International Conf. on Acoustics, Speech, and Signal Processing*. 2001. Salt Lake City, Utah

[17] Waibel, A., *et al.* "Advances in Meeting Recognition," in *Human Language Technologies Conf*. 2001. San Diego

[18] Winograd, T. and Guimbretiere, F. "Visual Instruments for an Interactive Mural," in *ACM SIGCHI CHI99 Extended Abstracts*. 1999. Pittsburgh, PA: pp. 234-235.