Multimodal User Interface Management

Boi Sletterink[1,2], Attila Medl[1], Ivan Marsic[1] and J.L. Flanagan[1]
[1]Rutgers University, CAIP Center, 96 Frelinghuysen Road, Piscataway, NJ 08854-8088, USA; Contact Email: medl@caip.rutgers.edu, Tel. +1-732-445-0080, Fax. +1-732-445-4775; [2]Delft University of Technology, KBS group

Objective and significance

The Rutgers University Center for Computer Aids for Industrial Productivity (CAIP) conducts research in human-computer communication. The objective of the STIMULATE project (Speech, Text, Image, and MULtimedia Advanced Technology Effort) within the CAIP center is to establish, quantify and evaluate techniques for designing synergistic combinations of human-machine communication modalities in the dimensions of sight, sound and touch in collaborative environments.

Human-computer interfaces have significantly advanced over the past decade. While most people still use only display, mouse and keyboard, alternative devices for human-computer communication, such as speech and force-feedback tactile interfaces, and gaze tracking have proven to be convenient and efficient to use for specific tasks (Cohen et al., 1996, and Flanagan and Marsic, 1997). Harnessing the potential power of these modalities, application developers may be able to increase the efficiency and accessibility of applications for people who do not use computers on a daily basis, as well as for the disabled. Hands-free interfaces are also of interest for applications in environments that do not allow the use conventional modalities, such as knowledge based systems for on-site technicians, emergency service personnel, and soldiers in the field.

We developed methods for fusing the modalities, using a speech recognizer and speech feedback system, a tactile glove and a gaze tracker in a military mission planning application. The prototype multimodal system is illustrated in Figure 1. Current work focuses on providing the same modalities to multiple applications in the collaborative environment, and providing an easy-to-use library for application developers. Providing these modalities to multiple applications means that they have to be managed, not unlike window managers for windowing systems. Therefore, we speak of multimodal user interface management.

Approach

We chose the DISCIPLE (Distributed System for Collaborative Information Processing and LEarning) framework (Marsic and Dorohonceanu, 1999) as our collaborative environment. The DISCIPLE framework is entirely written in Java, and therefore platform independent. It is run-time extendable because it runs Java Beans. The modalities we use are the Microsoft Whisper Speech system, the Rutgers Master II force-feedback tactile glove with gesture recognition software, and a gimbal-mounted gaze tracker from ISCAN, Inc. To make them

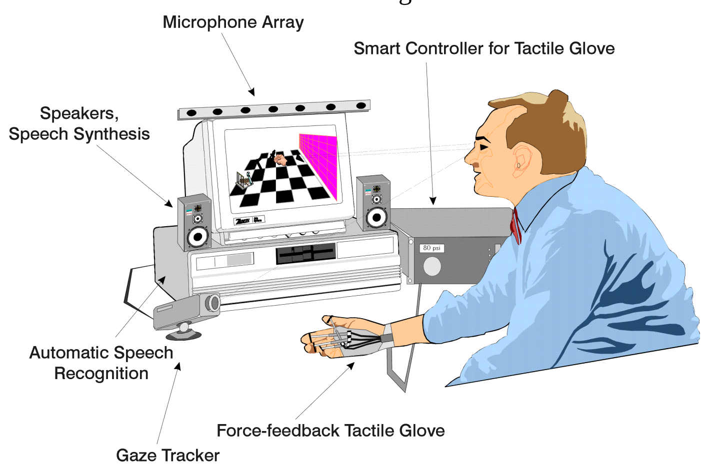

Figure 1: the Rutgers system for multimodal human-computer interaction

platform independent, pluggable and replaceable, they are connected to the manager by TCP connections.

The two main issues for the manager are routing messages between applications and modalities, and managing modality behavior and modality fusion behavior for different applications. To provide a base for building a manager, a communication interface between manager and applications was designed. This interface is to provide means for the application to find out what modalities are available, receive input events and send feedback through them.

The basic framework does not specify how the message (event) routing and fusion should be implemented. It only specifies methods for communication and introspection of capabilities and desired behavior, but it does not entail a specification for making decisions. This is up to the actual implementation, like with the X Windowing System, which has a variety of window managers, that all perform the same function and have the same interface but behave differently.

Implementation

Routing messages from modalities to applications is done through a notion of an active (selected or focussed) application. Only the active application will get modal input events. This means that, for example, recognized phrases from the speech recognizer will only be sent to the active application.

Routing messages from applications to modal outputs is implemented very straightforward: they are always sent to all outputs that they apply to. For example, textual feedback will be sent to both the on-screen dialog box and the text-to-speech system, if they are available at the time.

The other important issue, changing behavior of modalities according to the application that uses them, is also addressed by our system. The same notion of active applications is also applied to switching behavior: if another application is selected, the modalities are reconfigured. This means that for the speech recognizer, the recognition grammar is changed (allowing for more accurate recognition) and for the glove, the actions associated with the gestures are changed.

Fusion behavior may also change. For example, in a 3D application the glove does not perform mouse emulation, while in non-3D applications, it does.

Results

We have created a multimodal system that interacts with the user in the dimensions of sight, sound, and touch. In the collaborative desktop, several collaboration-aware applications were developed, including a whiteboard, a military mission planning extension for the whiteboard, and a medical image guided diagnosis system for the diagnosis of leukemia (Comaniciu et al., 1998). The military mission planning application allows the user to create and manipulate military units symbolized by icons on a terrain map (Medl et al., 1998).

We designed the modality management interface and implemented it as described above. This manager can be loaded on demand. Multimodal extensions have been added to the applications, and users are able to use any combination of these applications in the desktop. Adding multimodal extensions was straightforward thanks to the design of the multimodal interface.

References

P.R. Cohen, L. Chen, J. Clow, M. Johnston, D. McGee, J. Pittman and I. Smith, 1996. Quickset: A Multimodal Interface for the Distributed Interactive Simulation. In *Proceedings of the UIST'96 demonstration session,* Seattle.

J.L. Flanagan and I. Marsic, 1997. Issues in Measuring the Benefits of Multimodal Interfaces. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97),* Munich, Germany.

I. Marsic and B. Dorohonceanu, An Application Framework for Synchronous Collaboration using Java Beans, to appear in *Proceedings of the Hawai`i International Conference on System Sciences (HICSS-32)*, Maui, Hawai'i, January 5-8, 1999.

D. Comaniciu, P. Meer, D. Foran, A. Medl, Bimodal System for Interactive Indexing and Retrieval of Pathology Images, *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98),* Princeton, New Jersey, October 1998, 76-81. Also in demo session, 268-269.

A. Medl, I. Marsic, M. Andre, Y. Liang, A. Shaikh, G. Burdea, J. Wilder, C. Kulikowski and J. Flanagan, 1998. Multimodal Man-Machine Interface for Mission Planning, accepted for presentation at the *Intelligent Environments - AAAI Spring Symposium,* March 23-25, 1998, Stanford University, Stanford, CA.