

Exploring Handheld, Agent-based, Multimodal Collaboration

David R. McGee and Phil Cohen

Center for Human-Computer Communication,
Department of Computer Science and Engineering
Oregon Graduate Institute
P.O. Box 91000, Portland, Oregon 97291-1000
{dmcgee, pcohen}@cse.ogi.edu

ABSTRACT

Designing software for handheld computers is challenging because of their limited computing power, memory, display size, and the choices of hardware and operating systems. In addition to these challenges, collaborative handheld applications must deal with distributed computing and other network-related issues. We believe that by designing these applications as multimodal, multiagent systems, we can address some of these challenges. Taking this position, we have designed QuickSet, a multimodal application for group work on electronic maps. By leveraging QuickSet's agent-based roots, we were able to quickly add collaboration capabilities to it. In this paper, we will discuss QuickSet's collaboration capabilities, the ease with which they were added to the user interface, and plans for future extensions to these capabilities.

KEYWORDS: multimodal, cscw, groupware, handheld, collaboration, agents

INTRODUCTION

Handheld computing application design is challenging. First of all, handheld computers are limited in terms of computing power, memory, and display size. In addition to these constraints, collaborative applications compound these problems by their distributed, heterogeneous nature in wireless environments. However, the inherent mobility of handheld devices makes them poten-

tially ideal for certain collaborative work environments.

It is our position that several challenges of handheld devices:

- small screens
- slow processors
- the number of operating system and hardware choices

can be addressed by designing handheld collaborative applications as *multimodal* [1] (combining multiple, simultaneous input modalities, such as voice and gesture) and part of a multiagent system.

Multimodal interfaces

- scale well to both very large and very small display sizes

Multiagent architectures

- have a lightweight mechanism for application and interface interoperation
- are programming language and operating system independent
- provide a component-based software architecture that can distribute tasks to other agents on the network

Our research in the Center for Human-Computer Communication at the Oregon Graduate Institute and earlier work by Cohen, Oviatt, and colleagues at SRI International [2, 3] has focused on empirical and applied research in the areas of multiagent

architectures, multimodal human-computer interaction and human-computer collaboration. This research led to the development of the Open Agent Architecture (OAA)¹ [4] and QuickSet [5], an OAA-based application for interacting multimodally on maps, which runs on handheld to wall-sized PCs.

In this paper, we will present a description of QuickSet, its collaboration capabilities, lessons learned in its development that may begin to address some of the challenges identified above, and goals for extending these capabilities in the future.

1 QUICKSET

QuickSet is a tool for multimodal interaction with pen and voice on devices ranging from wireless, handheld computers to interactive wall-sized displays. With it, groups can work collaboratively on shared maps,² interacting with and controlling the maps, 3D virtual reality visualizations, simulation environments, and other “agentified” applications. The QuickSet collaborative user interface runs on a variety of computing hardware devices and operating systems from handheld or palmtop computers running Windows CETM to wall and table-sized displays running Windows NTTM or 95/98TM (Figure 1a and 1b).

The primary component of the QuickSet user interface is a geo-referenced map. Entities displayed on the map are registered to their positions on the terrain, and thereby to their positions on all user interfaces. The map provides pan and zoom capabilities, overlays, icons, etc. Using pen and voice together, the user can annotate the map, creating points, lines, and areas of various types, for example ditches, fortifications, berms, minefields, and swamps. The user can also create entities (i.e.,

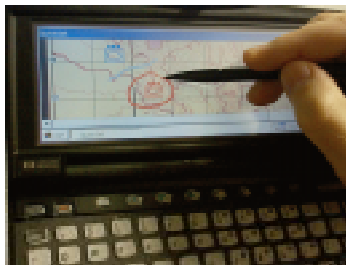


Figure 1a. QuickSet on an HP620LX running Windows CETM



Figure 1b. QuickSet on an ITI Digital Desk

vehicles or groups of vehicles), give them behavior, and watch a simulation unfold on the displays.

Though standard graphical user interface designs do not scale well down to handheld and palmtop personal computers or up to wall-sized displays, multimodal interfaces scale well regardless of display size. Since QuickSet can run on these size devices, the collaboration can range across them as well. In addition to its interface scalability, what makes QuickSet unique is that its digital ink is not only used for shared whiteboard-style interactions, but it can also combine with spoken input to produce a multimodal command, such as creating a new entity on the map. This style of human-computer interaction extends QuickSet’s collaboration capabilities far beyond that of a typical shared whiteboard application.

1.1 The Open Agent Architecture

QuickSet relies on a community of distributed artificial *agents*,³ based on the OAA, that work together to interpret and execute multimodal input (Figure 2).

Agents register their capabilities (the information that interests them and the requests for action that they are committed to performing) with a facilitator, which may be connected to a network of facilitators. Capabilities and messages, themselves, are in the form of Prolog terms. The facilitator unifies every message with each agent’s declared capabilities to determine whether an agent has registered an interest in the message. If so, the message is

¹ The Open Agent Architecture is a trademark of SRI International.

² Domains include, but are not limited to, distributed interactive simulation and military exercise initialization [6].

³ “Agents are active, persistent (software) components that perceive, reason, act, and communicate.[7]”

forwarded to that agent. The resulting framework can be viewed as a strategy for semantic or content-based multicasting [8]. This is not multicasting in the traditional sense, since the facilitator can act as a filter on the content, based on the capabilities registered and the binding of terms during unification.

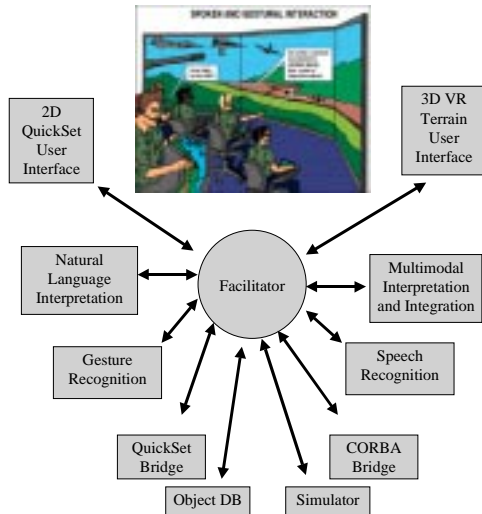


Figure 2. Open Agent Architecture

When two or more agents subscribe to and/or produce common messages, they (and their users) become part of a collaboration. The OAA offers a framework for heterogeneous collaboration, in that applications can operate on different types of hardware or operating systems, be written in different programming languages, and yet be part of a collaboration. For instance, when subscribing to the shared object database messages, user interfaces are notified of changes in the locations of entities, and render them in whatever form they choose, including 2-D map-based, web-based, and 3-D virtual reality displays. Consider this as a radical Model-View-Controller [9], where the model is managed by the object database agent, the view is managed by the user interface(s), and the controller is the suite of agents that can influence changes in the model by their action.

1.2 QuickSet Collaboration

Early in its development, QuickSet supported shared map objects. By relying on a distributed object database agent for the storage, manipulation,

and deletion of objects, different user interfaces could display the same objects. Recently, to allow for tighter synchronicity, we QuickSet to support the *coupling*⁴ of two or more user interfaces. When coupled, if one interface pans and zooms, the other coupled ones do as well. They do this by acting on and producing common messages. For example, coupled interfaces register a common capability to “ink” and they produce ink messages. Ink then becomes shared, immediately providing a shared drawing system. Users can interact with completely different interfaces, yet retain shared awareness. For example, colleagues at the Naval Research Laboratory have developed a 3-D VR terrain display that subscribes to and produces the same inking messages as QuickSet, thereby adding the ability to share ink with QuickSet.

In summary, the QuickSet user interfaces themselves are OAA agents. With agent messages, coupled QuickSets can share objects, views, and ink, much like a shared whiteboard. However, with QuickSet the ink can combine with spoken input to perform actions, such as creating objects or specifying orders.

2 LESSONS LEARNED

In a recent case study, Cohen [11] found our multimodal interface substantially out-performed a “state-of-the-art” graphical user interface for one expert in this domain using a standard 14-inch touch panel. A larger study will be underway shortly to confirm this finding. Our intuition tells us that smaller displays mean more compact representations and interaction mechanisms. There is nothing more compact than the spoken word when it comes to user interfaces. They don’t take up any space at all. Menus, dialogues, toolbars, and virtually every other form of direct manipulation can become cumbersome thieves of screen real-estate on handheld and palm-top devices.

For larger displays a similar argument can be made. In informal tests we have personally

⁴ Coupling [10] is defined as the sharing of tools, views, or other interface components.

experienced considerable fatigue while using a large interactive pen-activated display with a standard graphical user interface, which often forces the user to reach for the upper left corner of the screen. For this very same reason, designers of the Tivoli [12] system at Xerox chose to use a number of gestural commands in their interface. Large screens give Fitts' law [13] fits. We have found that speech combined with simple gestures can often meet user interface requirements for non-standard display sizes and this marriage seems especially ideal for synchronous shared whiteboard-style applications.

We are exploring the boundaries of using an agent architecture for handheld collaboration. The OAA was chosen for its flexible asynchronous, event-based capabilities, its ability to run the same set of software components in a variety of operating system and hardware configurations, and its easy connection to legacy applications. Additionally, the agent architecture supports mobility—less computationally-intensive agents (e.g., the map interface) can run on the handheld, while others (e.g., gesture recognition and natural language processing) operate elsewhere. Furthermore, based on the content of both messages and capabilities, the facilitator can be intelligent in its disposition, so that filtering can occur when multicasting a request to other agents. Finally, the same methods for inter-application communication are used for inter-widget (collaboration-based) communication.

However, the Open Agent Architecture's reliance on a highly flexible, string-based messaging, has its own limitations. It cannot efficiently transmit binary information, nor is it ideally suited for real-time, highly reactive information, such as cursor position or even shared ink, and furthermore it can become a communications bottleneck [8]. Consequently, one extension we have made to the agent architecture is the brokering of direct connections. Direction connections are a non-facilitated two-way communication link. We have built a prototype ink server agent using DCOM™. This server optimizes communication specifically for passing ink among collaborating agents quickly without paying the

penalty of marshalling and unmarshalling the data to and from an agent communication language.

3 FUTURE WORK

Current challenges in extending the collaboration capabilities of the QuickSet environment include: speech recognition on palm-tops, evaluating direct connections and other strategies for better synchronous behavior, adding session management, concurrency control, and rendezvous facilities, and finally supporting collaborative dialogue at the interface.

For multimodal applications to be truly useful on palmtop computers today, speech recognition must be off-loaded to agents on the wireless network. To support this, the palmtop's speech client agent will ask the facilitator to broker a direct connection to a speech recognition server. Similar strategies are currently being evaluated for real-time components, such as shared inking and cursors.

We are investigating different approaches to establishing and managing collaborative sessions. Currently QuickSet collaborations are established whenever facilitators are shared among interfaces. In the future, we will provide a session management agent, so that there can be a number of groups using the same facilitator, so that users can join specific groups. Furthermore, new versions of QuickSet have been designed as ActiveX™ controls; therefore they can be embedded in either web pages or documents (e.g., PowerPoint). We intend to establish whether or not it is feasible to embed QuickSet objects in mail messages in order to invite participants to join a collaboration.

A number of capabilities are being considered for development to better support synchronous collaboration in QuickSet, including agents for handling conflicting actions during a collaboration, including consistency checking and multi-user undo.

Finally, an effort is underway to investigate techniques so that QuickSet can be helpful when two or more people are working together on the same wall-sized electronic map. Currently QuickSet relies on a command-oriented dialogue to

take action. Evaluating collaborative multi-party dialogue to determine how a multimodal map-based interface can be helpful will be the emphasis of this thread of research. For example, QuickSet could highlight items as they are referred to in the discussion or moving entities when both people agree on its new location.

4 SUMMARY AND CONCLUSIONS

In this paper we have presented an overview of the collaboration capabilities of QuickSet, a handheld multimodal, agent-based architecture for map interaction, the lessons we have learned while building it, and plans for future handheld and groupware capabilities. Based on this experience we have adopted the position that designing handheld CSCW applications as agents in a multiagent system can address several problems inherent to these and related systems because of the open, heterogeneous, plug-and-play nature of multiagent architectures. Furthermore, small displays become less of a challenge when multimodal interfaces are adopted, since they scale to both small and large displays, unlike standard graphical user interfaces.

ACKNOWLEDGEMENTS

This work is supported in part by the Information Technology and Information Systems offices of DARPA under contract number DABT63-95-C-007, and in part by ONR grant number N00014-95-1-1164. Thanks to Josh Clow, Craig Minor, Joel Pothering and Ira Smith for their comments, encouragement, and continued support.

REFERENCES

- [1] R. Bolt, "Put-That-There: Voice and gesture at the graphics interface," *Computer Graphics*, vol. 14, pp. 262-270, 1980.
- [2] D. B. Moran, A. J. Cheyer, L. E. Julia, D. L. Martin, and S. Park, "Multimodal user interfaces in the Open Agent Architecture," In the proceedings of the 1997 International Conference on Intelligent User Interfaces (IUI97), Orlando, Florida, 1997.
- [3] S. L. Oviatt, "Multimodal interactive maps: Designing for human performance," *Human-Computer Interaction*, (special

issue on "Multimodal interfaces"), In press.

- [4] P. R. Cohen, A. Cheyer, M. Wang, and S. C. Baeg, "An open agent architecture," In the proceedings of AAAI Spring Symposium on Software Agents, 1994.
- [5] P. R. Cohen, M. Johnston, D. McGee, I. Smith, J. Pittman, L. Chen, and J. Clow, "Multimodal interaction for distributed interactive simulation," In the proceedings of Innovative Applications of Artificial Intelligence Conference, IAAI'97, Menlo Park, CA, 1997.
- [6] P. R. Cohen, M. Johnston, D. R. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow, "QuickSet: Multimodal interaction for distributed applications," In the proceedings of the Fifth Annual International Multimodal Conference, Seattle, WA, 1997.
- [7] *Readings in Agents*, M. N. Huhns and M. P. Singh, Eds. First ed. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1997.
- [8] D. L. Martin, A. J. Cheyer, and D. B. Moran, "Building distributed software systems with the open agent architecture," In the proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'98), Blackpool, Lancashire, UK, 1998.
- [9] Y.-P. Shan, "An event-driven model-view-controller framework for Smalltalk," In the proceedings of Object-oriented programming systems, languages and applications (OOPSLA'89), 1989.
- [10] P. Dewan and R. Choudhary, "A high-level and flexible framework for implementing multiuser user interfaces," *ACM Transactions on Information Systems*, vol. 10, pp. 345-380, 1992.
- [11] P. R. Cohen, M. Johnston, D. R. McGee, S. L. Oviatt, J. Clow, and I. Smith, "The efficiency of multimodal interaction: A case study," In the proceedings of International Conference on Spoken Language Processing, ICSLP98, Australia, 1998.
- [12] E. R. Pederson, K. McCall, T. P. Moran, and F. G. Halasz, "Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings," In the proceedings of InterCHI'93, 1993.
- [13] P. Fitts and J. Peterson, "Information capacity of Discrete Motor Responses," *Journal of Experimental Psychology*, vol. 67, pp. 103-112, 1964.