Single Display Groupware: Exploring Computer Support for Co-Present Collaboration

Lauren Bricker
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195, USA
bricker@cs.washington.edu

Kori M. Inkpen
School of Computing Science
Simon Fraser University
Burnaby BC, V5A 1S6, Canada
inkpen@cs.sfu.ca

Jason Stewart
Computer Science Department
University of New Mexico
Albuquerque, NM 87131, USA
jasons@cs.unm.edu

Brad Myers
Human Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891, USA
Brad.Myers@cs.cmu.edu

ABSTRACT

This panel will explore an interaction paradigm for co-present computer-based collaboration we term Single Display Groupware (SDG). SDG is a class of applications that support multiple simultaneous users interacting in a co-present environment on a single shared display with multiple input-devices. SDG are being used in various applications in the educational, entertainment and research communities, but many issues remain to be explored.

Keywords

Computer-supported cooperative work (CSCW); computer-supported collaborative learning (CSCL); computer-supported collaborative entertainment (CSCE), multiple input devices; single display groupware.

INTRODUCTION

Most computers found in homes, schools and workplaces are designed to accommodate one user's interactions at a time. Current research and development in CSCW and CSCL perpetuates this notion by focusing on the one-person/one-computer paradigm, facilitating collaboration through networking distributed workstations. By focusing all of our attention on supporting remote work, we have ignored a common form of group work that is in need of computer support. This panel will focus on Computer Supported Collaboration (CSC) from the perspective of enabling these co-present interactions.

For instance, when groups of kids want to collaborate on a computer they must often cluster around a small computer screen and share control of the computer via a single mouse and keyboard. Social and physical conflicts over the input devices often arise in these situations because current operating systems and application software only support one mouse and one keyboard. Similarly, in an office environment we are often restricted in our ability to collaborate seamlessly with others when that interaction involves a single computer.

We feel collaborative software should be designed to support people's natural interactions. For some tasks, people prefer to work or play together on the computer or they may be more productive as a result of this style of collaborative interaction [7]. However, the configuration of the equipment used can introduce barriers to successful collaboration. In a networked computing environment the collaborative effort may be unnaturally clumsy. Alternatively, collaborating around a personal computer may be socially uncomfortable for some users.

Single Display Groupware (SDG) is a class of applications that support multiple simultaneous users interacting in a co-present environment on a single shared display with multiple input-devices [11]. SDG allows users to interact more naturally and comfortably around the computer. Such applications take advantage of our human ability to interact and communicate in a face-to-face situation, and still benefit from computer technology.

Previous research in SDG includes Liveboard [5] and MMM [2] from Xerox PARC. Liveboard originally supported multiple cursors operating at the same time, but when produced commercially only supported one person with one cursor at a time. The Tivoli system [10] supports up to three people using pens simultaneously on the LiveBoard.

Recently, SDG has become a thrust in the educational and entertainment communities. In the learning domain, SDG creates new opportunities for peer learning/teaching, which may create a more effective learning environment. SDG has also been used in the entertainment domain in video games, both in arcades and for home use. However, SDG has met with some skeptecism in a work context in terms of its applicability to work-related tasks. With the introduction of multiple-input hardware systems (such as the Universal Serial Bus - USB) as a new standard on Pentium-based machines, and large wall-sized displays, support for SDG will be more widely available in the home and at work. We feel that it is an opportune time to explore SDG as an area of Computer-Supported Collaboration (CSC).

There are, however, many research issues that should be investigated to help design effective SDG applications. For instance, special care must be used when designing interactions between the users who must share access to public tools, such as menus and palettes. The default way in which these widgets and controls operate often make it inconvenient for multiple people to share them. For example, the typical pop-up menus when used by one user might obscure another user's actions. Programming techniques used to address these issues might be counter intuitive to the users, causing them to work in an unnatural or unexpected way.

This panel will be a forum for researchers to present and discuss issues relating to computer support for co-present collaboration. The goal of this panel is discuss the research issues regarding SDG in hopes of involving more people in this exciting and under-investigated area.

FORMAT

This panel will begin with a brief definition of SDG and a description of the research problem. Each panelist then will describe their research in SDG. Kori Inkpen will begin by describing her research, which gives strong motivation for the use of SDG and provides empirical results supporting the use of SDG. Lauren Bricker and Jason Stewart will present their systems that allow multiple users to work together on a desktop system by giving them each a separate mouse or pen-based input device. Brad Myers will present his work on supporting meeting room collaborations by enabling multiple PalmPilots to control applications on a single shared computer. Most of the presentations will include demonstrations of multiple people using the systems at the same time. Finally, we will open up the panel for a discussion to explore questions including:

- What learning, work, and entertainment tasks could benefit from collaboration on a single display? Are there certain tasks in each domain that lend themselves to specific SDG configurations? Where might SDG fail in each domain?
- What are the appropriate interaction techniques and user interfaces that can best be used by multiple people sharing a single display? What kinds of floor-control are appropriate?
- How can we overcome the hardware limitations of having multiple people interacting simultaneously? What requirements can we give designers of operating systems and application toolkits, so that SDG software can be more easily constructed?
- How can the research in SDG impact the design of future software and hardware?
- When is it appropriate to combine single-display groupware support with multi-display groupware support, and how should this be done?

We wish to involve the audience in this discussion to draw upon the expertise of the community for designing and developing collaborative applications.

POSITION STATEMENTS

Kori Inkpen

I will present observations and empirical findings from my doctoral research [6], which provides a strong motivation for exploration of SDG. I will address reasons why computer support for co-present collaboration is important; show how the one-person/one-computer paradigm is less effective in some instances; and provide results that demonstrate how adapting the computer environment to more effectively support collaborative interactions can bring about positive changes in motivation and productivity.

During the summer of 1992, together with other colleagues from the Electronic Games for Education in Math and Science (E-GEMS) research group, I observed people interacting with computer and video games at an interactive science museum. One of the most prominent observations during this two-month study was the degree to which people preferred to work with others while playing games, especially children. It was not uncommon to see three, four or five children clustered around a single computer playing for extended periods of time. Normally, all of the children were active participants, pointing to the screen, discussing ideas, and taking turns with the input device. Often the children appeared to be more successful as a result of this collaboration.

An empirical study was then conducted to compare children's attitude and performance while playing a commercial problem-solving computer game in one of three configurations: solo play (a single child playing alone); parallel play (a pair of children playing side-by-side, each with their own computer); integrated play (a pair of children playing together on a single computer). The results of this study showed that the children were more motivated to play and were more successful in the game when playing together on a single machine than in either of the other two conditions [7].

While playing together on a single machine was the most effective configuration in the previous study, sharing the mouse was often problematic. In a follow-up study, the computer was adapted to accept inputs from two mice. The software was not modified there was still only one cursor and only one mouse was active at a time. Control was transferred between the two mice using the right mouse button. Two turn-taking protocols were examined: *give* and *take*. Using the *give* protocol, the child in control of the game passed control to their partner by pressing their right mouse button. Using the *take* protocol, either child could take control of the game by pressing their right mouse button at any time. The results of this work showed that the addition of the second mouse did have an impact on the children's motivation and achievement in the puzzle solving game.

Kori Inkpen is an Assistant Professor at Simon Fraser University in British Columbia, Canada. She graduated with a PhD from the Department of Computer Science at the University of British Columbia in 1997. Her dissertation research focused on human-computer interaction issues for children's use of computers in educational environments, focusing on supporting children's co-present collaboration.

Lauren Bricker

I will present some of the activities implemented with Colt, a system that was developed as part of my doctoral research [3]. These activities will showcase a class of objects called *Cooperatively Controlled Objects* (CCOs). CCOs are designed to support simultaneous influences from two or more users on one or more machines. Furthermore, the degree to which the users must simultaneously manipulate a CCO may increase the amount and type of communication between the users [3]. CCOs can be employed to encourage users to focus on a task, a property that is especially beneficial when working with students.

Developing applications to support objects such as CCOs involves special challenges. An application designer must anticipate the interaction between the user and the computer, as well as the interactions between the users. Additionally, there is a lack of system support for development of software that accepts simultaneous input from multiple users. The Colt system is designed to address both of these issues. Colt is comprised of three parts: a design methodology, a software toolkit, and visualization and analysis tools. The design of the Colt system is motivated by a model of computer-supported collaboration that measures communication and cooperation between the users during an activity. A user study was conducted to illustrate and validate communication and analysis portions of the model. The user study showed that different types of interactions influence the type and frequency of user communication The study also showed that the users, when not restricted by the software to working together or apart, tend to work fairly closely together on a task. Finally, we found that users were not terribly frustrated by the enforced close interactions, and that they found the interaction entertaining.

The Colt design methodology aids developers in the creative process of planning a collaborative activity. The toolkit supports the implementation of collaborative programs by providing a shell application, a hierarchy of cooperative objects, and support for input from multiple users. Each cooperative object contains a mechanism to record a history of how users cooperatively controlled the objects. The visualization and analysis tools present different views of the object histories. These tools can be employed after program testing to see if the implementation meets the original design criteria.

Lauren Bricker graduated with a PhD from the Department of Computer Science and Engineering at the University of Washington. Her thesis entitled *Cooperatively Controlled Objects in Support of Collaboration* focuses on how simultaneous cooperative control of computer based objects can effect the communication between users in a collaborative activity.

Jason Stewart

I became convinced that technological support for co-present collaboration was an important and necessary research project while helping design and test single user applications in elementary school classrooms [4]. Kids wanted to be in control of the application. Having only a single input device was an irritating and artificial bottleneck to them. Although kids developed social protocols that enabled them to cope with the lack of collaborative support, it was clear from our studies that this was just a compromise on their part to put up with not being able to work in parallel.

Often the desired outcome of using a SDG application is not simply to complete a desired task, but more importantly, to complete the task in a collaborative fashion. For this reason, attention to the usability of SDG applications becomes foremost in their design. In the panel, I will share lessons learned from my experiences leading the iterative design and testing of KidPad, a SDG drawing program [10]. When designing SDG applications one must be aware of tradeoffs between functionality in the application and effective collaboration between partners. Because support of collaboration is often the primary goal, functionality may have to be sacrificed when it hinders collaboration.

Finally, I contend that we may be limiting ourselves by trying to force the WIMP (windows, icons, menus, and pointers) user interface metaphor to function in a multi-user paradigm. Exploring alternative approaches, such as tool based architectures [1], may lead to applications that are better suited for simultaneous use by multiple users.

Jason Stewart is a doctoral candidate in the Department of Computer Science at the University of New Mexico. His thesis, entitled *Single Display Groupware,* investigates how the use of SDG applications can enhance the collaborative behavior of kids working at a single computer. One focus of the work is the exploration of an alternative application architecture, the local tools architecture, to support building of SDG applications.

Brad A. Myers

Single-display groupware is not just for kids! In the Pebbles project (http://www.cs.cmu.edu/~pebbles), we are creating applications to connect multiple Personal Digital Assistants (PDAs) to a main computer such as a PC. We are using 3Com PalmPilots because they are starting to be ubiquitous.

There are certain kinds of meetings, including design reviews, brainstorming sessions, and organization meetings, where a PC is used to display slides or a current plan, and the people in attendance provide input. Many conference and presentation rooms today have built-in facilities for projecting a PC onto a large screen, and various inexpensive technologies are available for rooms that do not. When a PC is used as part of the presentation, often different people will want to take turns controlling the mouse and keyboard. For example, they might want to try out the system under consideration, to investigate different options, or to add their annotations. With standard setups, they will have to awkwardly swap places with the person at the PC. We observed that at most meetings and talks, attendees do not bring their laptops, probably because they are awkward and slow to set up, the batteries may not last long enough, and there is a social stigma against typing during meetings. Today, however, many people are taking notes on their PalmPilots. Therefore, we decided to investigate the use of PalmPilots to emulate the mouse and keyboard. The “Remote Commander” allows the users to take turns controlling the PC’s main cursor and keyboard so regular applications can be used, like PowerPoint or Excel. The Remote Commander has been downloaded over 3000 times in the first eight weeks since its release[1].

We also created a SDG shared drawing program, named “PebblesDraw,” that allows everyone to draw simultaneously. In creating PebblesDraw, we had to solve a number of interesting problems with the standard widgets such as selection handles, menubars and palettes. Palettes, such as those to show the current drawing mode or the current color, normally show the current mode by highlighting one of the items in the palette. This no longer works if there are multiple people with *different* modes. Furthermore, the conventional menubars at the top of the window or screen are difficult to use in multi-user situations for technical reasons, and also because they may pop up on top of a different user’s activities. The conventional way to identify different users is by assigning each a different color, but in a drawing program, each user might want to select what color is being used to draw, causing confusion between the color identifying the user, and the color with which the user will draw. We therefore developed a new set of interaction techniques to more effectively support SDG. These interaction techniques are implemented as part of the Amulet toolkit [9], by changing the behavior objects (called “Interactors”), the widgets, and the command objects to have a *User-ID* field that can be used to control how they are shared.

Brad A. Myers is a Senior Research Scientist in the Human-Computer Interaction Institute in the School of Computer Science at Carnegie Mellon University, where he is the principal investigator for many projects in the area of user interface software. He is the author or editor of over 160 publications, and is on the editorial board of five journals. Myers received a PhD in computer science at the University of Toronto where he developed the Peridot UIMS. He received the MS and BSc degrees from the Massachusetts Institute of Technology during which time he was a research intern at Xerox PARC. His research interests include User Interface Development Systems, user interfaces, Programming by Example, Visual Programming, interaction techniques, window management, and programming environments.

[1] The Remote Commander is available from the Pebbles web page at http://www.cs.cmu.edu/~pebbles.

REFERENCES

1. Bederson, B., et. al. Local Tools: An Alternative to Tool Palettes, in Proceedings of UIST '96, ACM Press, pp. 169-170.
2. Bier, E. and Freeman, S. MMM: A User Interface Architecture for Shared Editors on a Single Screen, in Proceedings of UIST’91, (Hilton Head, November 11-13, 1991), ACM Press, N.Y., pp 79-86.
3. Bricker, L. Cooperatively Controlled Objects in Support of Collaboration. Ph.D. Dissertation, University of Washington, Department of Computer Science and Engineering, Seattle, 1998.
4. Druin, A. et. al. KidPad: A Design Collaboration Between Children, Technologists, and Educators, in Proceedings of ACM CHI'97, ACM Press, pp. 463-470

5. Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, D., Pier, K., Tang, J., and Welch, B. Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration, in Proceedings of CHI'92, (Monterey, May 3-7, 1992), ACM Press, N.Y., pp. 599-607.

6. Inkpen, K. Adapting the Human-Computer Interface to Support Collaborative Learning Environments. PhD Dissertation, University of British Columbia, Department of Computer Science, August, 1997.

7. Inkpen, K., Booth, K. and Klawe, M. Playing together beats playing apart, espectially for girls, in Proceedings of CSCL'95 (Bloomington, October 17-20, 1995), ACM Press, N.Y., pp 177-181.

8. Inkpen, K., McGrenere, J., Booth, K., and Klawe, M. The effect of turn-taking protocols on children's learning in mouse-driven collaborative environments, in Proceedings of Graphics Interface '97 (Kelowna, B.C., May 1-23 , 1997), Morgan Kaufmann, San Francisco, pp 138-145.

9. Myers, B.A., et al., "The Amulet Environment: New Models for Effective User Interface Software Development." IEEE Transactions on Software Engineering, 1997. 23(6): pp. 347-365.

10. Pederson, E., et al. "Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings," in Proceedings INTERCHI'93: Human Factors in Computing Systems. 1993. Amsterdam, The Netherlands: pp. 391-398.

11. Stewart, J. et. al. When Two Hands are Better than One: Enhancing Collaboration with Single Display Groupware, in Adjunct Proceedings of CHI'98, ACM Press, N.Y., pp287-288.

12. Stewart, J. Single Display Groupware, in Adjunct Proceedings of CHI'97, ACM Press, N.Y., pp. 71-72.