Single Display Groupware

Benjamin B. Bederson*
Computer Science Dept. / UMIACS
Human-Computer Interaction Lab
University of Maryland
College Park, MD 20742
+1 301 405 2764
bederson@cs.umd.edu

Jason Stewart*
National Center For Genome Resources
1800-A Old Pecos Trail
Santa Fe, NM 87505
+1 505 995 4433
jes@ncgr.org

Allison Druin*
College of Education / UMIACS
Human-Computer Interaction Lab
University of Maryland
College Park, MD 20742
+1 301 405 7406
allisond@umiacs.umd.edu

Abstract

We discuss a model for supporting collaborative work between people that are physically close to each other. We call this model Single Display Groupware (SDG). In this paper, we describe the model, comparing it to more traditional remote collaboration. We describe the requirements that SDG places on computer technology, and our understanding of the benefits and costs of SDG systems. Finally, we describe a prototype SDG system that we built and the results of a usability test we ran with 60 elementary school children. Through participant observation, video analysis, program instrumentation, and an informal survey, we discovered that the SDG approach to collaboration has strong potential. Children overwhelmingly prefer two mice to one mouse when collaborating with other children. We identified several collaborative styles including a dominant partner, independent simultaneous use, a mentor/mentee relationship, and active collaboration.

Keywords:

Computer Supported Collaborative Work (CSCW), Human-Computer Interaction (HCI), Single Display Groupware (SDG), co-present collaboration, children, educational applications, input devices, Pad++, KidPad.

Introduction

We live in the age of the personal computer. The first personal computers were designed and built at Xerox PARC in the early 1970's with the fundamental assumption that only a single individual would be sitting in front of and interacting with them at any given time. This fundamental design legacy has carried on to nearly all modern computer systems. Although networks have enabled people to collaborate at a distance, the primary assumption still remains that only a single individual would need to access the display at a time. Therefore, computers have been by and large designed with a single mouse and keyboard for input, and a single visual display for output. Even the physical environments we place our computers in are typically designed for use by a single person: we often put our computers in the corners (Buxton, 1994) or put them tightly together so that there is only room for a single person.

> **Single Display Groupware**
>
> Computer programs which enable co-present users to collaborate via a shared computer with a single shared display and simultaneous use of multiple input devices.

In our work, we have investigated whether this single user assumption was a valid assumption or just a design legacy. In our day-to-day observations as researchers, co-workers, parents, and educators, we saw

* Much of this work was done when all three authors were at the University of New Mexico in Albuquerque, NM USA.

many times when people collaborated around computers. For example, while designing technology for elementary school children, we frequently observed two, three, and four children crowded around a computer screen each trying to interact with the computer application (Druin, 1999 ch. 3). Our research has also shown that children enjoyed their experiences with the computer more if they had control of the mouse and were actively controlling the application (Stewart, Raybourn, Bederson, & Druin, 1998; Benford et al., submitted). Our research has focused on trying to understand if the overall collaborative experience can be enhanced by enabling each partner to interact with the computer application independently.

We investigated how effectively current technology supports co-present collaboration by conducting a baseline study that used a commercial single user application in a collaborative setting and observing the collaborative behavior. We noticed a number of problematic behaviors that might be eliminated if each user had independent access to the computer. Given these indications, we began to investigate whether we could improve collaboration by explicitly building computers systems that support co-present collaboration as a fundamental property.

We found the collaborative experience of children was greatly enhanced by a simple conceptual change. We added multiple input devices to a single computer so that each child user could independently or collaboratively control the computer. We focused on the use of a single display because it most accurately reflects both what resources are commonly available as well as how computers are used today. Therefore, we have come to call this approach "Single Display Groupware" or SDG. Recent work including our own has begun to explore SDG, and in this paper we attempt to create a framework that ties together these different approaches, and motivates future system designers to include low-level support for SDG.

In this paper, we compare Single Display Groupware with other approaches that have been taken to support remote collaboration, and describe the unique requirements that co-presence imposes on the technology designer. We present descriptive studies that suggest existing computer technology is not well suited to support co-present collaboration, as well a number of studies which suggest SDG technology provides potential advantages for co-present collaboration. We describe a general purpose implementation architecture for testing SDG applications, and present a novel interaction technique called *local tools*, as a potential interaction metaphor for use in collaborative applications. Finally, we suggest operating systems modifications that we believe can offer opportunities for more varied, and better integrated SDG applications.

Vision of the future: The computer as a collaborative tool

The goal of our research has been to explore the potential of the computer as a collaborative tool. We began our work by visualizing how our interactions with computers would be different if they supported co-present collaboration. We realize, as others have, that computers will no longer stand alone, but must instead be viewed within the greater context of the environment within which they are used. Therefore, imagine in the not-so-distant future a computing environment where there is universal support for co-present collaboration:

Informal Collaboration (sharing computer at work)

At work, you are visiting the office of a co-worker to get feedback on your latest project. Since the Personal Data Assistant (PDA) you carry uses wireless networking technology, you can easily communicate with your co-worker’s computer. After she approves your log-on request, you start up your demo on her monitor, and use the touch screen of the PDA to control a cursor on her workstation. While she uses her workstation’s mouse to use your program, you gesture with your cursor indicating the areas you had questions about. As you expected, she finds a number of bugs in your code. But since you are both able to interact with the software, you work around the bugs without interrupting her or taking the input device out of her hand.

Marketing (exploration of interactive design)

At the designer’s office, you review the plans for the renovation of your living room. After going over some of the paper sketches, the designer offers to show you the 3D model of the renovation on his computer. He thinks it will give you a better idea of how his plans fit in with the rest of the house. As he guides the program into the living room, he encourages you to pick up the extra mouse and investigate the layout yourself. You have some trouble navigating with the unfamiliar software at first, but the designer demonstrates the navigation tools with his mouse and you quickly learn to mimic him. Together you both relocate furniture and experiment with different room layouts and color schemes.

Learning (exploration of science)

At school, your daughter is finishing work on her latest geometry project. She's having difficulty with proving the Pythagorean theorem and asks the teacher for help. The teacher is busy helping a group of students working at the other collaborative learning station, so your daughter's friend comes over to help. Her friend picks up one of the unused mice and together they explore the problem. They work together moving around the squares and triangles and measuring the results until they both feel more comfortable with the Pythagorean theorem.

Instruction (demonstration of new computer facility)

At the university, your student comes in to your office to show you the latest version of some software. When you get stuck, the student tries to explain how to use it. Feeling uncomfortable in taking the mouse from you, he notices the second mouse, and demonstrates how to solve the problem as you follow and then take over with your mouse.

Idea Generation (authoring tool)

At work, you are joined by two colleagues to brainstorm a plan for the newsletter. You all stand in front of an electronic "whiteboard" that enables you to draw with electronic pens. You start listing some content items for the newsletter on one side of the board while one of your colleagues starts sketching some possible designs on the other side. You work off each other's ideas – writing, talking, looking at what the other has written, smiling and giving subtle feedback. Your third colleague starts to get excited and picks up another pen to draw arrows between the sketches, showing the flow of ideas. Because this is electronic, you can clean up your work and save it as a starting point for future elaboration.

What's missing?

Although the field of Computer Supported Collaborative Work (CSCW) is thriving, and networked computing is one of the biggest selling points of computers today, the scenarios described above are not yet a part of today's world of computing. What is missing is computer support for *co-present* collaboration. The majority of research in CSCW today focuses on supporting people that are working apart from each other. Computers and networks are very well suited to supporting remote collaboration, but supporting people that are working together requires solutions to new problems.

Based on the computer paradigm discussed in this paper, Single Display Groupware (SDG), we suggest an increase in effort that investigates technology that brings people together for "shoulder-to-shoulder" collaboration that enhances the interaction of people working together in one location.

Alternatives to a single display

Our research has focused on a restricted subset of the possible solutions for supporting co-present collaboration. It is important to understand that our approach is one among a growing number of researchers' in this emerging area of the field of Computer Supported Collaborative Work (CSCW).

Co-Present Groupware

It could be argued that we have severely limited ourselves by only investigating solutions that involve a single display. We could have chosen to expand the scope of our model to include multiple output devices, and called it Co-Present Groupware (CPG). However, the goal of this work was to study the architectural concerns that arise while supporting multi-user collaboration around a single Personal Computer (PC). The overwhelming majority of current PC systems provide only a single display for output. Most schools are heavily resource-limited and aren't likely to purchase specialized collaborative learning hardware. In addition, while collaboration is important in the work environment, a majority of office workers do a significant portion of their work independently and are likely to continue to do so even if a superior collaborative technology was introduced tomorrow. Therefore, any technological solution that requires specialized hardware is likely to exclude a significant fraction of the user groups that are most likely to benefit from those systems. We have therefore chosen a more restrictive path to first explore how far co-present collaboration can be used with existing hardware before a redesign of computer hardware is suggested.

Networked Groupware in side-by-side format

Most researchers have studied the use of networked groupware systems in settings in which partners were either physically remote from one another, or at least out of sight of one another. However, researchers have studied the advantages of enabling co-present collaboration by using networked groupware in a side-by-side configuration. By using systems that were designed for traditional distributed groupware, and it can be technically easier to explore SDG. For instance, two computers can be set up in a tightly coupled shared mode (e.g., with a shared whiteboard application). Then, the mouse from the second computer is put physically next to the first computer, and both users look at the monitor from the first computer. Then, from the user's perspective, they are using the same kind of SDG system described in this paper even though it is implemented with multiple computers. This has been done in a few systems, such as with the Klump application using the DIVE CSCW system (Benford et al., submitted).

Other approaches to co-present collaboration

Besides our own work there have been a number of other groups that have explored the problem of supporting co-present collaboration.

Navigational systems

There are some notable existing systems that use co-present groupware today. Aircraft cockpits and videogames are excellent examples of the kind of systems we are proposing. However, these systems only touch the surface of what is possible. Their principle limitation is that they are largely navigational systems, and don't provide support for authoring. In addition, their interfaces are mostly in hardware, where each user has their own specialized hardware input device with one physical manipulator for every action you can perform. So, while we can learn from these systems, there is much more to be done for software-based SDG systems.

Aircraft Navigation

A number of vehicles including aircraft and drivers-education training cars have shared interfaces that control the vehicle. Once again, few interesting SDG design decisions can be explored using these systems. To begin with, all the interfaces are in hardware. In addition, the users of these systems have rigid predefined roles that govern how and when they have access to the shared controls (e.g., the pilot out-ranks the co-pilot, and the instructor out-ranks the student). Added to this, users of these systems are not capable of independent activity unless they have separate hardware controls that the other user doesn't have. For example, the co-pilot may have navigation instruments that the pilot doesn't have. The steering wheels in the drivers education car each control the same low-level hardware (the wheels, the gas flow, and the brakes), the instructor, however, is able to over-ride the student in an emergency. In an aircraft, both navigation sticks move identically when one is moved, and social conventions determine who holds the stick at any one time.

Video Games

There are other examples of technological support for co-present collaboration that we place in the category of hardware interfaces. The most significant may be multi-player video games. While these are software-based, they primarily support users navigating through scenes and shooting things, playing ball, or fighting. They do not support shared creation of information. Aside from spatial navigation, they do not support a great deal of information retrieval. Therefore, while the social issues of video games are interesting to SDG designers, they do not offer us as much guidance for interface development as one may initially think.

Early collaborative systems

Several research projects explored the use of specialized computer technology that enabled collaboration within a single room.

Shared desks

An early experiment aimed to understand the needs of co-present collaboration was performed at Xerox PARC (Olson, 1989 ch. 8). They built special "corner desks" that were designed for two people to sit around a single computer. They put these desks in the corners of conference rooms and public spaces, and found that they were never used. While putting the desks in storage, one researcher decided to put one in his office, and the ensuing months found himself frequently sitting at the desk with another person just as they were originally designed for. This experience led these researchers to conclude that the layout and positioning of these kinds of systems are crucial – if they are not where people naturally work, they won't get used.

Shared rooms

The CoLab project, like other electronic meeting rooms, provided each member with a desktop computer which allowed private work as well as control of a shared display at the front of the room (Moran et al., 1997). Earlier shared rooms were built by Krueger as installation art pieces (Krueger, 1991). One drawback of electronic collaborative rooms is that they require expensive, specialized hardware that is prohibitive to many people who could benefit from enhanced support for co-present collaboration, for example school children.

Digital Whiteboards

The Liveboard digital whiteboard (Stefik et al., 1987) and the Tivoli application enabled multiple simultaneous users (both co-present and remote) to interact with the shared digital whiteboard. The authors point out that simultaneous use of the whiteboard rarely occurred and they speculated that the lack of adequate software level support for co-present collaboration (of the kind presented in this paper) may have been the cause.

Single Display Groupware systems

There have also been a number of other researchers that have begun to explore some of the technical and social issues involved in building SDG software.

Architectures

An early implementation of SDG was MMM (Bier & Freeman, 1991). It enabled multiple co-present users to interact with multiple editors on the same computer display by providing each user with an independent input device. The system was never made available to the research community, and no user studies were conducted to investigate the limitations of the idea. MMM was not pursued, but some of the researchers working on it transferred this technology to study the use of multi-handed input for single users (Stone et al., 1994).

Using the Colt architecture, Bricker built SDG applications that teach collaborative skills (Bricker, 1998). The guiding metaphor of applications built with her SDG architecture is the 3-legged race: the goal is not to enable participants to run the race faster than they could individually, but instead to require participants to learn to cooperate in order to be able to run at all. Example applications include a color-matcher in which three users must find the RGB values for a given color, and a chord matcher where users find the notes for a given chord.

Learning

Other researchers have investigated how SDG technology could influence groups in a learning environment. Work by Inkpen (Inkpen et al., 1997) showed that by providing each user with a separate input device gave significant learning improvements, even when only one device could be active at a time. The active device could be toggled through a predetermined access protocol. This is an important result because it indicates that SDG could benefit tasks in which both users are not expected to work simultaneously, such as editing a paper.

Personal Digital Assistants

The Pebbles project (Myers et al., 1998) investigates the use of hand-held Personal Digital Assistants (PDAs) as portable input devices in an SDG setting. Early work explored how multiple PDAs could be used together with existing software in a meeting environment. Lack of software support meant that only a single individual could interact at any given time, but each user had their own input device, so control could be easily transferred by social protocol. Later work explored explicit software support of multiple PDAs, as well as how an existing GUI application toolkit, Amulet (Myers et al., 1997), could be modified to support SDG.

Public and Private Workspaces

Rekimoto also developed a multi-device approach, which enabled users to create work on a palmtop computer and then move the data onto a shared public computer, such as a digital whiteboard. He called this the "Pick and Drop" protocol (Rekimoto, 1998). Work by Greenberg and Boyle has also been investigating the boundaries between public and private work by designing applications that can be used collaboratively in both an SDG setting using PDAs or over a network using a workstation (Greenberg & Boyle, 1998).

Models to help understand SDG applications

To better understand the implications that SDG will have on computer system design, we need to investigate how SDG applications differ from other applications. We discuss these differences using the Model-View-Controller design introduced by the Smalltalk community, and then focus on a description of the I/O channels that computers use.

Model-View Controller (MVC)

The Model-View-Controller (MVC) approach of the Smalltalk community provides a way to illustrate the differences between SDG and other groupware systems. The *model* corresponds to the underlying information of the program, the (i.e., the data). The *view* corresponds to the part which controls the output channels of the system, while the *controller* corresponds to the part that handles the input. Traditional groupware systems have a single shared model, and since each user has a separate computer, each has a separate view-controller pair that communicates with the shared model. SDG systems also have a single shared model, but differ from traditional groupware systems by only having a single shared view through which the computer must give feedback to all users, and a single shared controller through which all users interact with the computer. SDG applications could have multiple controllers if an application wanted to replicate all user interface elements and provide every user with a unique copy (Stewart et al., 1999). This solution seems unlikely to scale as it would quickly take up all available screen space for the user interface.

Channels

User Interfaces consist of *input channels*—which enable users to communicate with the computer, and *output channels*—which enable the computer to communicate with its users.

We define an input channel to be an input device that provides *independent* input to the computer (Stewart et al., 1999). So for example, in current computer systems the mouse and the keyboard would not be considered separate input channels since the keyboard input is dependent upon the mouse for setting keyboard focus. Future computer systems may support an independent mouse and keyboard but current ones do not, so the *typical* current system will be described as having only a single input channel. In some cases, such as laptop computers, there can be multiple pointing devices, *i.e.*, an external mouse and a trackpad. These devices are also dependent and share the same input channel—either both share control of the system cursor, or only one can be active at a time. This definition covers the observation that dividing up tasks by giving one user the mouse and another the keyboard is not likely to result in a good collaborative experience (Papert, 1996 p. 89).

We define an output channel as a part of the computer interface that uses an independent modality to provide user feedback (Stewart et al., 1999). Examples would be a display for visual feedback, speakers for audio feedback, and a force-feedback joystick for haptic feedback. Most current computers have the potential of using both visual and audio feedback, but most UIs use little or no audio feedback and rely almost exclusively on visual feedback. There are exceptions to this, such as audio systems for blind users, but these are in the overwhelming minority of existing systems. This could change with future systems, but the *typical* current system will be described as providing a single output channel.

Characteristics of SDG applications

Now that we've proposed models for examining the characteristics of SDG, what are the consequences of these characteristics in terms of actually building applications?

Shared Screen Space

In the design of user interfaces, there is a general tension between maximizing the functionality of the program and maximizing the amount of screen space available for user data. This trade-off becomes more apparent for SDG applications. Since there typically must be controls to manage each user, there may be even less screen space available for data. This can result in the need for larger displays for SDG applications.

Coupled Navigation

There is a general issue of how to manage navigation through data. Using MVC terminology, we can say that whenever one user navigates to a different part of the Model the other users will be affected. If the coupling is tight, then all users will navigate together when one navigates. If the coupling is loose, then other users may have part of their Views obscured by one user navigating to a different area of the Model.

Shared User Interface

Even though users have separate input devices, the interface elements through which the user communicates with the computer (menus, palettes, buttons, etc.) must be designed to handle multiple simultaneous users. This restriction corresponds to the single shared Controller in the MVC description. This has a direct impact on the design of SDG applications. For minimum functionality, most interface elements can be locked so they can only be used by a single user at a time. For enhanced functionality, new mechanisms that support simultaneous use must be developed.

Shared Feedback

The interface elements through which the computer communicates state information to users (buttons, palettes, etc.) will likewise be shared by all users and must be capable of relaying information to all users simultaneously. This is a consequence of the shared View from the MVC discussion. This means that interfaces that depict global state (such as current pen color) must be redesigned to accommodate state per user.

Shoulder-to-Shoulder Interaction

People work differently side-by-side than they do at a distance. Subtle non-verbal cues that people give, consciously and unconsciously are ubiquitous – and we are all very good at picking up on these cues. By supporting people working together shoulder-to-shoulder, SDG systems can take advantage of these fundamental human qualities (Ishii et al., 1994; Smith et al., 1989; Hall, 1966 pp. 108-111).

We will discuss each of these issues in more detail throughout the paper with the exception of shared navigation. Exploration of the shared navigation problem is very important but very difficult. In the work described herein, navigation issues were explored only rudimentarily.

What interaction techniques work in SDG?

Many interface widgets and interaction techniques, including menus, palettes, button bars, scrollbars, etc., were developed with only a single user in mind. However, when two or more users are interacting with those widgets simultaneously, the interaction model can break down. This section explores some of the difficulties in trying to utilize these techniques in multi-user applications.

Interaction Semantics

We define the issue outlined above as the study of interaction semantics. The general problem with interaction semantics is*: how should widgets developed for single user interfaces function in a multi-user environment*? What do the users expect to happen? In our design of the KidPad application (Druin et al., 1997) we discovered that there was often not a single answer, and that user would often become confused as to how interaction should function in a collaborative user interface.

As a case in point, let us consider the problem of selection handles. If an object is selected by a single user, all of that object's eight selection handles will be the identical color and shape. In this case the semantics are clear: the user who has selected the object can interact with any of the handles. A user may be permitted to interact with the handles of another user or he may not. That is an application dependent decision to rely on explicit exclusion or implicit social protocols. A more confounding example is a multi-user undo tool. If a user accidentally erases an object created by a different user, who would undo the mistake? The user who created the object, or the user that accidentally erased the object? While testing a prototype tool, different users expected different behaviors, and considered it a software error if it the tool behaved differently than expected.

But what happens when two users have a common object selected, as is illustrated in Figure 1? The user with circular handles has selected all three shapes, and the user with spade-like handles has only selected the 5-pointed star. Some of the handles belong to one user and some to another and one in the lower left corner is shared. In this case the semantics may be hard to predict for a user, because the application has made it appear that some handles are different than others.

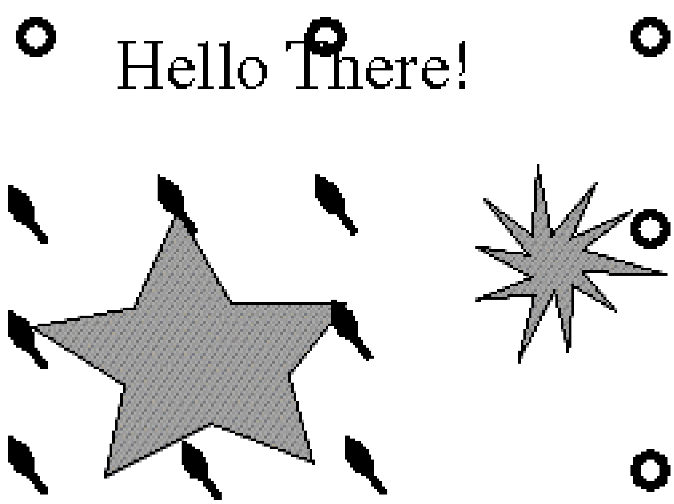

Figure 1: The multiple selection problem

The problems with interaction semantics are not limited to selection. What happens when two users interact with different parts of a scrollbar? How should a menubar react when multiple users click on different menus, or how should a single menu respond when opened by one user, but another user chooses a selection? Many widgets that have obvious functionality in single user interfaces may be inappropriate in multi-user applications. This will require an evaluation of existing user interface techniques and metaphors to see which are appropriate to use in SDG applications.

A solution to this issue might be using explicit locking protocols to govern when users can and cannot interact with either widgets or data objects. Myers has investigated the use of two different protocols to govern widget interaction: *one-at-a-time widgets* and *anyone-mixed-together widgets* (Myers et al., 1998). In their architecture, scrollbars and menus were implemented as one-at-a-time widgets: any user could initiate an interaction, but until that user finished interacting, the widget would only accept input from that user. This meant that when any user activated a drop-down menu, all menus were inactivated for all other users until the first user was finished. Canvases were an example of a widget that allowed anyone-mixed-together interaction: all users could create and modify graphical objects simultaneously. The tradeoffs between locking (one-at-a-time interaction) and mixed interaction will be discussed in more detail in the next section.

Social protocol versus technological constraint

In many of the preceding examples, the semantics of the interface elements in multi-user applications was not clear. It might be possible to prevent conflict by explicit technological constraints, such as locking widget interaction to a single user until that user has completed interaction. This explicit control may not be necessary and may in fact be detrimental. For example, in the case of selection handles, if no locking is present and any user may interact with another's selection handles, then the interaction semantics become much clearer. In many cases social protocols may suffice for avoiding direct conflict.

There may be instances where accidental interference might best be explicitly prohibited. In the Pebbles system (Myers et al., 1998), simultaneous interaction with traditional widgets such as pull-down menus are managed by only allowing one user to interact with a given widget at a time. For example, as soon as one user presses on a menu option, the other users are not allowed to interact with that menu until the first user releases their mouse. Specifically, the other users are locked out.

Alternatively, in some traditional groupware applications users have chosen to remove exclusion constraints and allowed social protocols to govern their interactions (Greenberg & Boyle, 1998; Shu & Flowers, 1992). It is likely that there are user groups who would perform better when governed by explicit exclusion control. For instance, groups that are competitive, and are not always trying to support one another. The most flexible solution may prove to provide both mechanisms and allow users to chose between them when necessary.

Local Tools and SDG KidPad

To understand the ideas surrounding SDG, we built a computer architecture for supporting SDG called the Local Tools architecture (Bederson et al., 1996; Stewart et al., 1998; Stewart et al., 1999), and a test application called KidPad using it. The Local Tools architecture is unique in that it was built to support SDG applications as a fundamental property.

The Local Tools architecture is described in detail elsewhere (Stewart, 1998), but briefly is structured as follows. Traditional interface widgets, such as pull-down menus and tool palettes, are replaced with "local tools" that are displayed on the screen co-located with the data. All functionality is represented as individual tools. To use a tool, one clicks on it to pick it up, and then clicks anywhere else to use it. Figure 2 shows some of the tools used in KidPad.

These local tools work well in an SDG environment because they eliminate the need for global interface state. Instead, each tool has its own state. For instance, instead of having a global pen color, each crayon tool has its own color. In this way, if a user wants to change crayon color, s/he changes the color of the tool s/he is using without affecting any other user's tools. Note that this design does not have any state per user, only per tool. So instead of setting a foreground color that would affect all the tools a particular user accesses, this only changes the color of that tool.

Figure 2: Some tools used in KidPad: An eraser, a bomb (for erasing an entire drawing), some crayons, and a hand (for moving objects).

Trade-offs of local tools

Tools can be configured once and reused. So if a user needs two tools of similar function each can be pre-configured and the user can swap back and forth without needing to reconfigure. Because tools can be placed along side the data, they can be configured, and left where they are needed.

The primary difficulty in using local tools is that the tool themselves must be managed. Because each tool sits on the data surface, they use up screen space. To help manage this, we developed the notion of toolboxes which are used to put tools within. We have a few toolboxes where each box holds a specific set of tools. Clicking on a closed toolbox opens it up so that its tools are positioned neatly on the screen. Clicking on an open toolbox closes it, hiding all the tools inside the toolbox.

We developed the interface concept of local tools specifically for use by children in SDG applications. They cleanly avoid many of the problems of simultaneous use of more traditional interface widgets, and simultaneously offer a very physical model for user interaction. We aimed for this physical model because many of the young children we initially worked with had difficulty learning the abstract ideas behind traditional interface mechanisms such as pull-down menus and tool palettes.

Implementation

The Local Tools architecture described in this paper was implemented for Linux/X and runs on standard Pentium class PC's without any special hardware. Extra input devices (mice and tablets) are plugged into existing serial input ports. Up to three simultaneous devices have been used with this approach.

Due to bugs in early XFree86 implementations of the X graphics system on Linux[1], Local Tools required that only one instance of any single device type was used. This required support for a device type other than mice in order to enable multiple devices. We chose to use tablets (Wacom art-PadII, model KT-0405-R). Tablets provide a rich feature set that matched well with the KidPad drawing program.

[1] The XInput library that supports "Input Extension" events had made an assumption that only one device of each type would be used. It was implemented with a global variable holding *the* device's position. We worked with the developers of that module, and were able to get it fixed for the final evaluations of KidPad. This fix enabled any number of devices of a given type (e.g. mouse, tablet, joystick, etc.) to be used simultaneously.

The X input system is typical in that while it does support multiple input devices, it does so through a special mechanism. It distinguishes between the primary, or "core", devices and secondary, or "extension" devices. This approach to operating system support of devices adds a substantial burden to the developer of SDG applications. The users of these systems don't consider one mice to be more special than another, and so the application developer must hide the technical difference between the devices from the user. Ideally, future operating systems would support multiple input devices in a consistent manner.

Baseline Study: How do children use existing technology?

We chose to focus on children for our initial exploration of SDG applications. This was for several reasons: children are a group that is likely to benefit from SDG because they often work at computers in groups of up to four individuals (Strommen, 1994; Berkovitz, 1994; Druin & Solomon, 1996); also, many of the tasks that children do using computers could be augmented by enabling each child to interact with the commuter at the same time (Schneider, 1996).

Children are an ever-growing group of technology users, who are demanding and economically important to the computer market (Heller, 1998). At the same time, children are also representative of novice computer users in general, and design ideas learned from children are often applicable to novice adult users (Milligan & Murdock, 1996). However, compared to novice adult users, we have found that children are more adventurous when attacking new problems (Druin, 1999). Today's children are growing up in a computer generation that is not intimidated by technology. Adult novice users, on the other hand, are more likely to be cautious and intimidated by the technology. In addition, children are not as concerned with social niceties such as politeness when criticizing a design – they are direct (Druin & Solomon, 1996). If it is bad they will let it be known either with words, body language, or both. Children can also be very demanding users – children expect to have fun while doing everything, while adults have learned to expect dullness and boredom, especially in computer interfaces (Soloway, 1996).

Before building new SDG technology it was important to determine how effectively existing single user technology supports co-present collaboration, and to characterize any shortcomings exhibited by existing systems. Two formative studies were conducted to investigate how effectively single user applications supported a group drawing task. The first study used a commercial drawing application for kids, KidPix. The second study used a prototype SDG drawing application, but provided each group with only a single mouse in order to simulate the single user condition. This second study served as a control to allow comparison between the baseline study and later studies of SDG applications. This section will describe the results of the study and what implications it has for the design of SDG technology.

Baseline study methodology

We worked with 72 New Mexico elementary school students, ages 8 – 12 years old, for 40 sessions over a period of 3 months. Students participated in the study during an after-school program which gave them the use of the school's computer lab. Since the computer lab was only one part of the after-school program it was difficult to randomly select participants for the study. Each day we worked with pairs of users currently present in the lab who had not yet participated in the study.

It should be noted that all user studies were paired in same-sex groups. There are a number of potential challenges that have been indicated by previous researchers when using mixed-sex groups (see for example (Grossmann, 1995 p. 27) and (Inkpen, 1997)). Since it was not the goal of this study to determine the effects of gender on collaboration, we made the decision to eliminate this from our study.

Every pair of students participated once in the baseline study. Students were asked to collaboratively create a drawing of a playroom. They were instructed that they would be given about 10 minutes to draw their playroom using the commercially available KidPix program, and then they would be asked to tell a story about the playroom afterwards. We participated alongside the students, asking questions, and taking notes. One of the authors (Stewart) observed all user groups and recorded most groups on videotape for future reference.[2]

[2] Due to technical problems, only 28 of the 40 groups observed were recorded and scored.

Baseline study results

There was a large difference in the behavior observed for the active (mouse-controlling) user and the passive (non mouse-controlling) user. In order to better understand the patterns of activity, we calculated frequencies of behavior. One of the authors (Stewart) watched the recorded sessions and counted occurrences of each behavior, and averaging the total number of occurrences over the 10-minute session. Three categories of behavior were scored:

1. Verbal communication:
   - Commands expressed by active/passive partner.
   - Queries expressed by active/passive partner.
   - Frustration expressed by active/passive partner.
2. Non-verbal:
   - Pointing at screen by active/passive partner.
   - Use of keyboard by passive partner.
   - Attempt to take control of mouse by passive partner.
3. Attention:
   - Attention of passive partner wanders to non-task activity.

The most frequently occurring behaviors were found to be:

- The non-mouse user often pointed at the screen: In an attempt to physically manipulate screen objects and interact more directly with the application, the non-mouse users tried to make their desires known by indicating to the mouse user what they would like to do. There was an average of 8 hand pointing events every 10 minutes.
- Users fought for control of the input device: Every 10 minutes, there were an average of 5 attempts to physically take control of the partner's mouse. These occurrences ranged in severity from both partners always keeping their hands near the mouse to having one partner physically restrain the other.
- There was user frustration: The non-mouse user expressed irritation over not being an equal participant in the collaborative effort an average of once every 10 minutes.
- The quality of communication was not collaborative: Most talk was from the passive partner issuing orders to the active partner. Only 6 of the 28 groups scored showed behavior where the active partner actively solicited opinions from the passive partner.
- There was lack of attention: the non-mouse user tired of watching and looked away from the task an average of 3 times every 10 minutes.

These differences centered on a single theme: the non-mouse partner desired to maintain involvement in the task by having some level of control of the application.

We were concerned that for the baseline study, a comparison of the KidPix application with an SDG application would introduce some confusion. KidPix has such a rich set of professional features such as sound, pre-drawn stamp objects, and fun visual effects while our anticipated SDG application would offer some of these features, but not all. Therefore a second study was performed in which 12 pairs of children used a prototype SDG drawing application, with only a single input device (thus simulating a typical single user application). The children used the application for 15 minutes and were asked to draw any picture of their choosing (they were given a 5 minute warm-up period to familiarize themselves with the program, followed by a 10 minute drawing phase). The results observed using the SDG prototype were not qualitatively different than those using KidPix – all classes of behaviors were observed in both cases in similar proportions. Therefore it appears to be fair to compare the baseline study using KidPix to future studies using SDG applications.

Baseline study conclusions

The results of the baseline studies suggest that using existing single user technology in a co-present collaborative setting can lead to unwanted conflicts or tension because partners have unequal control over the application and an unequal participation in the task. Other studies have demonstrated findings similar in nature to this study – that single user systems can be used to support co-present collaboration, but the explicit technology development to support co-present collaboration would be likely to improve collaborative behavior (Mateas et al., 1996; Inkpen, 1997; Vered, 1998).

Pilot Studies with SDG technology

Before building a complex SDG architecture, a number of small pilot studies were conducted using prototype SDG applications to explore whether group interaction gave any indication of being augmented using SDG technology. These pilot studies used early prototypes of the KidPad technology that we continued to develop for use in the final study.

Formative Study 1

The first study was conducted as part of a Technology Workout at CHIKids during the ACM CHI 97 conference. Technology Workouts were short (3 – 4 hour) periods where kids used experimental computer technology and gave feedback to the designers. Twelve children of varying ages (6 – 9) were given the opportunity to use both a fully developed single user drawing application for kids (as described in (Druin et al., 1997)), or a prototype of a SDG drawing application. The prototype supported two children, one using a mouse and the other using a drawing tablet. Each input device drew in a pre-defined color, either red or blue, and each device could erase all lines it had drawn. The interface had two zoom buttons, zoom-out and a zoom-in, that smoothly changed the scale of the drawing. There were also save and reload buttons, for the children to save drawings for later printing. All data was collected by participant observation, with two of the authors (Bederson and Stewart) observing.

Children enjoyed the diversity of tools in the single user application, but the kids were most excited about using the prototype SDG application. This was a surprising result for us, as we had expected the rudimentary interface of the prototype SDG application would limit the children's enjoyment, but instead, they lined up to be able to use it with their friends. Also, we witnessed a number of new behaviors that we had not seen while attempting co-present collaboration using single user applications. First, the children seemed to have a great deal more fun using the SDG drawing application. One pair of girls used the prototype for over 45 minutes jumping up and down, singing and dancing. When the screen would get too full of lines they would erase their work and start again. Another pair created an interactive story by one child continually zooming out or in while the other drew a circle around the center, creating a continuously growing or shrinking spiral and telling a story about space travel. One boy worked by himself using the prototype to create a dynamic story. He would use one device to draw a boy, and the other device to draw a cage around the boy. Then while other kids watched he would tell the story about the boy who got locked up in the cage, but was so strong that he could break free and escape. When he came to the part about breaking free, he would click the delete button on the device that had drawn the cage, making it vanish, leaving the boy uncaged.

Formative Study 2

A more formal follow-up to the above study involved 24 children from Hawthorne Elementary school in Albuquerque, NM. They tested a more advanced prototype of the KidPad program. They worked in 12 groups using the application for 15 minutes to draw a picture of their choosing (a 5 minute warm-up to familiarize themselves with the application followed by a 10 minute drawing phase). Compared with the baseline study using KidPix, the kids using KidPad exhibited a higher attention to the task, less frustration, less pointing at the screen, and less command-oriented communication. Data was gathered by participant observation with one of the authors (Stewart) observing. Also, sessions were recorded on video tape and scored using the same instrument from the baseline study. The instrument from the baseline study was used in order to enable as much comparison between the two studies as possible, but it did not work as well as we had hoped. That instrument focused on behavior differences between active (mouse-using) and a passive (non-mouse-using) partners. In this study, however, both partners were active, and so less information was gathered and many of our conclusions are given as qualitative only.

It was also noted that:

- The students seemed to have more fun, smiled more, and looked at each other more often. Even the two students who said they liked KidPix better never let their attention wander away from the activity.
- Curiosity and exploring the interface was enhanced: users swapped input devices, learned to use tools together, erased each other's work, tried to use multiple devices on one tablet, and even tried four-handed manipulation of a single input device.
- Kids were aided in their ability to find things, because they could do it together – so if a task is problem-solving or creative in nature, SDG supports collaboration, which in turn supports the task.

Pilot study conclusions

These results suggest that SDG technology may be well suited for use in co-present collaboration. Some novel behaviors were observed when children used SDG technology to collaborate at the same computer display, including peer-teaching, curiosity, and having fun.

Descriptive study

This section discusses the methodology of a field study we conducted one year later using a significantly enhanced version of KidPad to evaluate what collaborative changes can occur from the use of an SDG application.

Study participants

Sixty students, ages 9 – 11 years old, from the Hawthorne Elementary school in Albuquerque, NM participated in this descriptive study. They were grouped into pairs and were randomly assigned one of two conditions, using KidPad in either a two input device condition or a single input device condition. Each group worked on the project during four separate 15 minute sessions during their regularly scheduled computer class time over a period of one month.

Study experience

The children were instructed to collaboratively create a series of drawings which they would enter as a team into a design contest sponsored by the University of New Mexico. They were told that the contest was being held because we were creating technology for kids and we wanted to know what kids thought about technology and what they thought computers of the future should be like.

The first session was a warm-up session that gave students a chance to interact with KidPad and become familiar with it. During the second session they were told that the contest had started and that they should work together as a team to create two drawings, one that showed what computers of the future should be like, and the second that showed what they wanted to do over the summer (the study was conducted during the final month of school and summer vacation was on the children's minds).

During the fourth and final session, each group had their conditions switched – if they had been using KidPad with two devices they were only given one device, and vice versa. They were then given the task of creating a final drawing that showed where they would go if they could travel anywhere (some groups chose to travel within the US, others to other continents, some chose to travel through space, and one group traveled back in time).

Study application

This section describes the application that was evaluated, and what subset of the available local tools were used. A number of different tools were investigated during the design of KidPad. Not all of these tools were used during the study. A number of tools had been superseded by better tools, and a number of tools were not fully implemented by the time the study was begun. The interface that the kids used during the study is shown in Figure 3. There were two hand tools (one for each user), four crayons (two fat tips and two thin tips), one color tool (with 10 predefined colors), one eraser, one bomb, one grow tool, one shrink tool, and a scrapbook.

- The hand tool allows picking up and moving of drawn picture objects.
- The crayons were simple tools that created a line directly under the crayon tip when pressed. The color and width of the line drawn matched the visual representation of the tool.
- The color tool showed 10 color swatches, and clicking on the color tool with a crayon would change the color of that crayon for future use.
- The eraser tool had two modes: a rubout mode and a click mode. The rubout mode is entered by clicking and dragging the eraser over a series of objects. While in rubout mode, all objects drawn by the user of the eraser are erased, and objects drawn by the other users are left untouched. Click mode is activated by clicking on an individual drawn object. In click mode, any object can be erased regardless of whom created it. The differential security in the two modes was created in order to protect partners work from accidental deletion.
- The bomb tool will delete all drawn objects created by the user of the bomb tool. Once again the security was added to protect accidental erasing of one partner's objects by the other partner.
- The scrapbook stored the user's saved pictures. Users can load a previously saved drawing into the current drawing by first clicking on the scrapbook to show available drawings, and then clicking on a drawing to load it into the current one. After being loaded, a drawing can be moved as a unit using a hand tool, or resized using the shrink or grow tools. Drawings are saved into the scrapbook by first clicking on the scrapbook, and then choosing the "Save" entry. Using this mechanism, the children created their own clip art, first creating a simple drawing, saving it into the scrapbook, and then loading it into another drawing as many times as desired.
- The shrink and grow tools can be used to make either individual objects or entire drawings smaller or larger respectively.

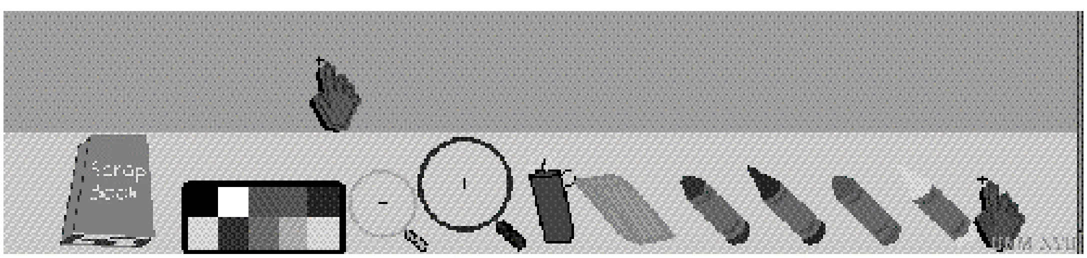

Figure 3: User Interface of KidPad During the Study

Data collection

We collected data for this study using the following five mechanisms:

- Drawings created by the groups: The users could save any drawing to the Scrapbook for later use. A total of 153 drawings were saved.
- Participant observation: One of the authors (Stewart) observed every group that used the program, answering their questions about the task, the interface, and interacted with them socially. We also took notes of their interaction, and occasionally asked them questions about the interface, or the task.
- Video Recording: Most of the 104 sessions were recorded onto 8mm video tape for later review. Technical problems resulted in 5 missed sessions. Content analysis of the video tapes was performed using the instrument given in Table 2.
- Program instrumentation: We wrote the software to provide logging functionality at the level of tool interaction. The following events were logged: when tools were picked-up, dropped, or swapped

(one tool picked-up while another was dropped); when tools were activated by pressing a mouse button, deactivated by releasing a mouse button, any motion, and any dragging (motion while a tool is active). Each event was recorded with a time stamp, which device and which user caused the event, the current tool location, and what object was affected by the event.

- Informal Survey: After the final session, the teams were given an informal verbal debriefing, to see how they felt about their ability to work together as a group in each condition. They were asked which condition they felt was the easiest to do drawings, which condition was the most fun, and which condition they would choose for use in other applications.

Results of study

Informal survey of children

Due to scheduling difficulties only 23 of the 30 groups were able to complete the final session[3]. We anticipated that the groups would be split as to which environment they considered the easiest to use, either single or multiple input devices. However, only 7 children (15%) thought that one device was easiest to complete the drawings, while 37 (80%) felt the two device condition was easiest, and 2 children (4%) were undecided. 45 children (98%) answered that they felt that it was most fun using two devices. Only one child (2%) thought that one device was more fun. The answers to the question of which condition kids would like to use for other computer applications were identical to the answers for the question of which condition was most fun. This suggests that having fun may be more important for kids than efficiency of task completion.

The children were also given the opportunity to say why they felt either condition was better. The one girl who preferred the one-input device condition did not say why. The others described why they preferred the SDG condition. The summary of the most frequent responses is in Table 1.

Response	Frequency	Examples
No turn taking	49% (81)	"We didn't have to share"
Parallel work	35% (16)	"We can do different stuff at the same time"

Table 1: Results of informal debriefing. Frequencies are shown in percentages and then actual occurrence account in parenthesis.

In response to our question of why they preferred SDG, one child commented "because there's two mouses!" (many of the kids thought it was obvious that two had to be better than just one). Another said "if [my partner was stuck and] I wanted to help there's another mouse" (peer-teaching was an advantage that even the kids were aware of). One girl said "[with two mice] you could do whatever you want" (KidPad did not enforce collaboration, children could work individually if they chose).

The majority of children (77% (20)) who had used the two mouse condition complained loudly when they were only given a single mouse for the final session: "Hey! Where's the other mouse?" and "If there's only one mouse, I'm going back to work at my other computer" were common reactions. The opposite reaction was common in groups that had only used a single mouse and were now given two mice: "Coool!" was the nearly unanimous response (90% (18)). One girl, when initially chosen to be involved in the study, refused to participate. She had worked previously with us during the baseline study and was frustrated over having to share. When told she didn't need to share anymore because there were two input devices, her attitude changed completely, and she participated in all four sessions.

[3] The realities of working in a school environment offered us challenges. One of which was absenteeism. Since it was so close to summer, absenteeism was higher than normal – if one partner skipped school, the group was given a makeup date. Some groups couldn't complete all their makeups.

Video tape analysis

One of the authors (Stewart) reviewed each of the 104 video recorded sessions and performed a content analysis of the recordings using the instrument provided in Table 2. The instrument is split into two identical halves, one for the left partner, and the other for the right partner. Each half is split into eight sections:

- **ADRUNF:** These categorized the type of verbal statements made by the users. Each of the six letters correspond to the following choices: Agreement, Disagreement, Rude, Unknown, Narrator comment (made by the person running the study), Follow-up (user responds to narrator).
- **CSQ:** These indicate the strength of the verbal statement, either a Command ("Put that over there!"), a Statement ("Let's put this over there."), or a Query ("Where should we put this?").
- **TIO:** Classifies whether the statement applies to the Task (refers to the picture being drawn), the Interface ("How do you use the bomb?"), or Other (gossip with partner or participant observer).
- **Other:** Often used this to write down examples of what was said.
- **NV:** This was for Non-Verbal communication, for example we would often indicate laughs, giggles, smiles, and other indications of fun here.
- **PV:** This was for Para-Verbal communication, for example, we would indicate anger, irritation, or surprise here.
- **Role:** This would indicate which of the four roles the partners were working under, Single, Independent, Mentor/Mentee, or Collaborative.
- **Computer Action:** This was used to indicate when partners changed tools or used a given tool (for example, the scrapbook or color tools, which couldn't be picked up).

Each line of the instrument would then indicate a particular action by one partner or less frequently, a joint partner action. This instrument was developed in collaboration with researchers from the University of New Mexico's Communications Department, and the College of Education.

Partner 1								Partner 2							
ADR/UNF	CSQ	TIO	Other	NV	PV	Role	Computer Action	ADR/UNF	CSQ	TIO	Other	NV	PV	Role	Computer Action

Table 2: The instrument used to collect videotape content analysis data

Discussion of Collaborative Styles

In analyzing the initial data, we found that one could say there were four major styles of collaborative interaction during the study:

- **Domineering:** This was the case when one partner was disinterested in the task and allowed the other partner to dominate, providing little or no input or feedback. The other partner tended to take over the session and basically do everything themselves.
- **Independent:** This was the case when each partner had a different idea of how to complete the task. Partners with only a single input device would often decide that one partner would complete his drawing while the other watched and then they would switch roles. Partners with two devices would sometimes draw a line down the middle of the screen (sometimes literally using the crayon tool or figuratively with a partner saying "this is my side … and this is your side") and each would draw a separate picture.
- **Mentor/Mentee:** One user teaches the other. This often occurred when children were attempting to use the interface, although sometimes individuals with a great deal of artistic creativity would demonstrate different drawing styles and techniques to their partners.

- **Collaborative:** This happened when partners would first talk about the task and how they would approach it, then divide the task up in some way, and then give feedback to one another while they were working.

It is important to note that the partners' collaborative style could change a number of times during a session. For example, partners might enter a mentor/mentee mode for only 30 seconds or so while one partner demonstrated the use of a particular tool. Also, partners could spend part of their time doing separate drawings as two independent designers and then decide it would be more effective to work collaboratively. For example, one group changed back and forth between a collaborative style and working independently four times during a single session. The frequencies of the styles observed during the study are given in Table 3.

Condition	**Designer Styles**			
	Domineering	*Independent*	*Mentor*	*Collaborative*
One Device	8	26	0	16
Two Devices	1	38	4	24

Table 3: Frequency of Collaborative Styles

The most frequent styles were independent and collaborative. It is interesting to note that, the ratio of independent to collaborative is nearly the same in both conditions (0.62 and 0.63). This suggests that the addition of a second input device didn't cause a dramatic shift from individualistic behavior to collaborative behavior. The two noticeable differences are the lack of obvious mentoring in the single input device condition, and the low amount of domineering use in the two input device condition.

Mentoring was one of the significant differences observed in comparison to the previous pilot tests of early SDG technology. It was surprising to see only four occurrences in the study compared to the higher frequency observed during the pilot studies. One explanation for this may be the relative simplicity of the interface used for the study so that users quickly became adept at using all tools. If the application had allowed more complicated interaction perhaps more mentoring would have occurred.

One potential reason for the low frequency of domineering use in the two input device condition is that users appear less bored or apathetic when they each have an input device. By having an input device, users always had the potential for interaction with the computer and their partner (whether they chose to use it or not). That potential for interaction could also be one explanation for Inkpen's findings that the presence of two input devices increased the learning even when only one device was active at a time (Inkpen, 1997).

It was sometimes difficult to assign a classification to the activity of groups with one input device. Because only one partner could interact at a time, behavior would often overlap between domineering and independent use, or between collaborative and independent use. Often, one partner would draw a complete picture, save it, and then pass the mouse to the other partner to draw. There was not a clear line that separated the behaviors, so often a group would be scored as exhibiting both the domineering and independent styles.

The lone example of the domineering style in the two device condition was rather unique. The one partner did 95% of the drawing and communicating while the other partner helped draw, but usually erased whatever she drew shortly afterwards. The main user would often prod her partner with comments like, "Hey, c'mon Mo, you gotta help me out!". In that group's final session with only a single input device, the domineering style was also exhibited. The one partner would finish drawing a piece of the picture, hand the mouse over to her partner, and say: "C'mon Mo, you draw yourself!", but her partner would refuse, pushing the mouse away. We believe that this illustrates that the change in technology from single user applications to SDG applications is not a magic bullet that will always transform groups that collaborate poorly into groups that collaborate well.

Discussion of Final Experiences

The final session for each group was a switched condition session. Groups which had previously been in the single input device condition were given two input devices and vice versa. A majority of groups switching to the multiple device condition expressed excitement, and a majority of groups switching to a single device condition expressed disappointment. Besides these general reactions, A number of specific descriptive observations were made from the video analysis of the final sessions.

Only one user (Brittany) indicated that she thought that using a single input device was more fun, but she didn't indicate why during the debriefing. She and her partner were in the two input device condition, so in the final session they were only given a single input device. During the two input device sessions, Brittany was observed to sometimes interfere with the progress of the drawing, erasing what her partner had drawn. During the single input device session, Brittany completely controlled the interaction. She voluntarily gave up the mouse on six occasions during the session (usually due to verbal prodding by her partner), but she would forcibly take back the mouse after only a few seconds on each occasion. It seems that she enjoyed the single input device condition more because it enabled her to dominate the interaction. In the two device condition, the most she could do was interfere with her partner's work.

There were a number of positive changes observed when groups in the single input device condition were given two input devices. For example, one group that acted timidly with the application during the single device condition interacted in a much more confident manner during the two-device session. During their sessions with only one device, both partners seemed very self-conscious of their drawing skills and would frequently erase what they had done, passing the mouse to their partner saying, "You draw it". However, during the final session with two input devices both partners were much more playful in their interaction and less self conscious. They each drew separate stories simultaneously – one drew Ohio, while the other drew Mars. Perhaps it was because neither partner was focusing their total attention on what the other was doing (since each was drawing a separate story) they were more relaxed and less self conscious.

Another group that had a positive transition from the single input device condition to multiple input devices seemed to do so because of a difference in interaction styles. In this group, one partner, Reylynn, verbally dominated the interaction. She would chatter and giggle at very odd and seemingly inappropriate times. She would also break into fits of giggling when her partner made a mistake, and would often take the mouse away from her partner on those occasions. This made it uncomfortable for her partner to draw, because she would be criticized whenever she made a mistake. During the final session with two input devices, Reylynn would still giggle when her partner made a mistake, and she even once took her partner's mouse away, even though she had her own mouse. However, because her partner was working on her own part of the drawing, using her own input device, she seemed less bothered by Reylynn's behavior and actually enjoyed herself.

The most significant positive change from single device to multiple devices occurred with Gary and Devyn. With only a single device, each was bored when the other was drawing, often looking away from the computer to talk with an adult researcher, fiddle with a neighboring computer, or just looking around the room. The final session with two mice was as if two different children were at work. They laughed frequently, and had a high degree of interaction with one another. They paid attention to the task for the duration of the session, never looking away a single time. Even though they had used the program three times previously, they began exploring the interface in ways they hadn't done before seeing how tools worked and seeing if they could interfere with one another. They successfully demonstrated one of the unique collaborative advantages of SDG applications over single user applications: users can interact with another, not just the application. This made dynamic interaction and dynamic collaboration possible. They told "moving" stories, almost like simple puppet shows with KidPad, sometimes erasing each other's work, sometimes moving it around, talking and telling stories the whole time.

There were also some groups whose collaborative behavior worsened when changing from the multiple device condition to the single device condition. For example, one partner, Crystal, would do most of the drawing and talking, even though she frequently encouraged her partner, Maureen, to be an equal contributor. Even though Maureen did little work, she did participate and offer ideas and suggestions, albeit infrequently. It appeared she felt self conscious of her drawing ability, as she would often draw something then erase it immediately afterwards. Crystal rarely used the eraser, even though she would laugh at how poorly she had drawn something. In the final session with only a single input device, Maureen did not give any real feedback or offer ideas, and she refused to draw even when Crystal put the mouse in front of her and asked her to complete a part of the drawing. It appears that Maureen's self consciousness became even greater when there was only a single input device.

There was one noticeable group whose collaboration improved when it switched from the multiple device condition to the single device condition. In this group, Shawn appeared to have a learning disability. During the sessions with two input devices, Shawn's partner, Marty, would often be absorbed in his own drawing and wouldn't communicate much with Shawn. During these sessions, Shawn would often stop to watch what Marty did and would then lethargically draw something similar to what Marty drew. However, during the final session with only a single device, Marty and Shawn communicated a great deal about what they would draw, and Marty frequently handed the mouse to Shawn and asked him to draw a part of the story. Their work was much more collaborative, and Shawn played a much larger role in the work.

Patterns Over Time

It was difficult to categorize behavior changes in the groups over time. Because the first session for every group was mainly an exploration session, and because the final session was the switched condition session, it only allowed two sessions to observe behavior changes. One significant observation was that "scribble wars", which were a frequent occurrence in the pilot studies of SDG applications, were limited to the initial warm-up session of the study. In later sessions, although scribbling occurred, it was self-regulating. If one partner started to scribble and "mess up" the picture, the other would often complain, and the scribbling partner would stop. It was hoped that by having a design contest in which the partners would be entering their drawings as a team would motivate the teams to care about their work. This could have reduced the amount of scribbling, or the novelty of two mice with one screen could have been a factor as well.

Discussion of Collaboration Change

For many groups, being able to work simultaneously required a new skill that needed to be explored before it could be effectively utilized. For example, Carrie and Virginia initially had difficulty working together with two devices. They each attempted to draw a different picture and interfered with one another's work. At one point Carrie said, "I don't like this, kinda like working together. I kinda like working by myself." But just 5 minutes later the team had begun to have a lot of fun together, discovering what could be done when both interacted at the same time, and Carrie said, "That's better! We should work like a team like this."

Other groups naturally used two devices. Denae and Alicia had two very different work styles. Denae was patient, methodical, never used the eraser, and laughed at her own mistakes. Alicia was nervous, changed tools often, often erased what she drew immediately after drawing it, and had periods of inactivity. With two mice they always drew separate pictures. With one mouse they attempted to work on the same picture, but Denae's calm, patient style conflicted with Alicia's nervous style. They each laughed at the other's mistakes and chided each other, but the collaboration worked because they were obviously friends. Humorously, in the final debriefing when they were asked which condition was best they both answered simultaneously, each giving a different reason why two devices was best. They agreed however, that the main advantage was "Because we both get to draw our own pictures." This is an example of how having two devices helped users with very different work styles interact more effectively.

The groups that were classified as independent showed different qualities depending on whether they were in the single or multiple device condition. For example, Ashley and Aricelia used two devices and although they would each work on their own drawing, they frequently communicated with one another and laughed a great deal about their ideas and drawings. Gary and Devyn used a single device and would often draw a complete picture and hand the device to the other. The partner who was not currently drawing would often look around the room, fiddle with the computer next to him if it was unoccupied, or chat with the participant observer. As indicated earlier, they were one of the groups that showed the biggest positive collaborative change from one device to two devices. With two devices they both paid attention to the task for the entire session. They interacted with each other's drawings creating dynamic stories, and had lots of fun.

There were exceptions to these general classifications. Iliana and Christina were in the one device condition, and they were one of the most collaborative groups. They always discussed what they would draw with one another, and frequently switched the single mouse back and forth. When switching to two devices they encountered a number of initial difficulties that illustrate why designing SDG interfaces is more complicated than developing single user interfaces. Iliana imported a drawing from the scrapbook, and Christina wanted to remove it with the bomb, but she was unable to because the bomb only eliminates objects 'created' by the bomb's user.[4] They also tried to swap tools that the other partner was currently using, and they also got visually disoriented trying to figure out which partner was controlling which tool. But after a short time of confusion, they quickly got used to the interface with multiple devices. When asked which condition was best, they both enthusiastically answered the two device condition. Iliana said that she wanted every computer to have two devices. They didn't like having to take turns even though they were very good collaborators when they took turns. Even more interestingly, they pointed out that by only having a single mouse it took more time to complete a drawing. With two devices they could work in parallel and accomplish more: "We got this whole planet done in this much time!"

Conclusion of behavior analysis

During the month long descriptive study, a number of situations were observed that indicate how SDG technology can be superior to the use of traditional single user technology in co-present collaboration. Also, a number of situations were observed in which SDG technology creates new problems that single user technology didn't have. Also, as could be expected, SDG did not appear to be a magic bullet. Some groups in the SDG setting performed very poorly together, so the mere presence of one input device per user is not sufficient to transform all collaboration into good collaboration. These findings suggest that SDG may not be appropriate in all situations and that more detailed descriptive studies are necessary in order to better understand when and where SDG technology is best suited for the task.

Analysis of Automated Data Logs

Instrumentation routines were inserted into KidPad that logged information to disk whenever the KidPad tools were used. We logged a timestamp, which user was responsible for the event, which tool was in use, what kind of action it was (motion, drag, drop, swap, button-press, or button-release), the x and y coordinates of the event, and what object was effected by the event. A total of 70 megabytes of data was logged during the month long KidPad study.

The event log confirms our analysis of the videotape in showing that when two devices were simultaneously available, each device was in fact being actively used. Table 4 summarizes the event quantity data. It shows that there were close to twice as many of each event type when two devices were available as compared with one.

	Single Device	**Two Devices**
Total Events	6,539 (1,941)	11,592 (5,571)
Motion Events	4,557 (1,423)	7,702 (3,690)
Swap Tool Events	25 (10)	47 (20)

Table 4: Comparison of event quantities by single and two devices. Numbers are listed as *average (standard deviation)*.

Another difference that was observed in the use of KidPad between the single device and the multiple device groups was the complexity of their drawings. Table 5 summarizes the information about user's drawings. The largest difference was that the average number of story objects in each drawing is nearly twice that for the single device groups. Also, the two device groups used 50% more colors in their drawings than the single device groups did. As was mentioned earlier, the users noticed that they were able to draw more complicated pictures when they each had a separate input device.

[4] The iterative design of the bomb and eraser interface was a long process. Later addition of the scrapbook required a decision of what protection the imported objects should have: they could maintain the creator that originally drew them, they could all be given the creator who imported them, or they could be given both, allowing either partner to erase them. The children decided that the person who imported the drawing should be the creator. As the case with Christina and Iliana showed, it may have been the most popular solution, but the interaction semantics were not necessarily obvious to a new user.

	Saved Drawings	Objects	Colors
Single Device	2.95	36.15	3.79
Two Devices	2.85	71.73	5.04

Table 5: Summary of drawing statistics. Data is averaged over all groups in each condition.

Discussion and Future Directions

Tradeoffs of the SDG approach

A primary focus of our work is to see collaborative computer tools examined within the context of their use. We foresee that the final outcome of the use of any SDG technology is therefore likely to depend heavily on the individuals in question, and the context of their collaboration, so it is of limited value to make sweeping generalizations of when and where the technology will be useful. So, while our experience is related to the specific tools we have built and users we have worked with, we present a summary of the issues we have seen, noting that not all of these will apply in all situations.

Potential Advantages of the SDG Approach

Current computer systems do little to encourage collaboration of multiple users. Single user systems provide only one explicit input channel for all users, so if multiple users attempt to collaborate using such a system it is up to the users to develop a sharing mechanism for utilizing that channel. In contrast, SDG applications will have an inherent notion of multiple co-present users and will provide each user with an equivalent input channel. This could have an impact on many aspects of using computers together. Some possible benefits are as follows:

- *Enabling collaboration* that was previously inhibited by social barriers. For example, in many cultures there is often a reluctance to invade the personal space of another person. The personal space surrounding close friends is smaller than that surrounding co-workers and acquaintances, and the space surrounding strangers is the largest of the three (Hall, 1966 Chapter X). Due to these proximate effects, many people may be inhibited from attempting to share a computer when another person is sitting in front of it. By explicitly providing for a separate input channel, the personal space around the person may be decreased enough to allow another person to comfortably interact with the computer at the same time.
- *Enabling types of interaction* that require multiple users. Bricker has explored a number of collaborative interactions that require multiple simultaneous users at a single computer. The goal of her research was to create tools that would strengthen collaborative learning (Bricker, 1998). In our current work (Benford et al., submitted), we have been investigating encouraging collaboration by adding special features to the tools that are activated only when the children work together.
- *Enriching existing collaboration* at a computer. For example, turn-taking is often viewed as unnecessary and cumbersome (Shu & Flowers, 1992). Enabling multiple input devices will in some cases enable work to be done in parallel, making the collaboration both more efficient and more enjoyable in the eyes of the users (Stewart et al., 1998; Druin et al., 1997). Also, a number of studies have indicated the benefit of shoulder-to-shoulder collaboration due to the collaborators enhanced verbal and nonverbal communications (Smith et al., 1989; Hall, 1966 p. 108–111).
- *Reducing or eliminating conflict* when multiple users attempt to interact with a single application. Often it is difficult to create an appropriate sharing mechanism for the shared channels, or it is difficult to obey the mechanism created (Stewart et al., 1998). By providing separate channels, potential conflicts are pushed one step further away.
- *Encouraging peer-learning and peer-teaching*. When existing single user technology is used in a collaborative learning setting, the competition between users to interact with the application can inhibit the learning benefits of collaboration (Stewart et al., 1998). By providing applications with multiple communication channels, it is possible to enrich learning by diminishing competition for access to the input channels (Papert, 1996 p. 89).

- *Strengthening communication skills*. Because strong-willed users can no longer monopolize a task by merely controlling the input device, users may have to communicate more with each other to resolve conflicts.

Potential Disadvantages

New conflicts and frustrations may arise between users when they attempt simultaneous incompatible actions. Working in parallel can be an advantage, but it can also be a disadvantage if users each have separate conflicting agendas. One serious concern in this area is navigation. Since there is only a single shared output channel (the display), if one user decides to navigate elsewhere in the data space, it may negatively affect the other users since all users' navigation is coupled.

There are two extremes of this coupling: loose coupling and tight coupling. If that coupling is tight (for example, two users working with a shared text editor containing a single text window) then if one user navigates to a different part of the space, then the other users will also navigate there as well. If the coupling is loose (for example, a racing car video game that splits the screen down the middle giving each player a separate view) then when one user navigates, the other users may not navigate there, but they may have their work occluded, or they may be distracted.

We have identified a number of potential solutions to the coupled navigation issue:

- Sharing the single screen, and using social protocol to determine who is allowed to navigate and when.
- Sharing the single screen but using a locking mechanism that prevents a user from navigating when another user is working.
- Using dynamic views to provide each user a temporary decoupled view when that user navigates.
- Providing each user a separate view by splitting the screen into one area per user.
- Not allowing any navigation.

Each of these solutions carries both advantages and disadvantages. After the option of not allowing any navigation, the option requiring social protocol to control navigation is the easiest to implement, but it enables every user to navigate even if it's against the desires of the other partners. Using a locking mechanism can eliminate the problem of one partner navigating against the others' desires, but users have often indicated that use of such locking mechanisms can inhibit the flow of interaction (Shu & Flowers, 1992). Also, implementing a mechanism with clear semantics that is less frustrating than using social protocol may be difficult.

Using dynamic views that are created when one user navigates, and eliminated when users re-enter the same viewing area might be a nice middle ground between tight and loose coupling, but they are likely to be the most difficult to implement in such a way that the semantics for creation and elimination of the temporary views is clear to the users. Current multi-player video games almost exclusively use the option of splitting the screen into one view per user. This option does not scale well beyond two users as it causes a dramatic reduction in the amount of screen space available to each user. Also, it may isolate the partners from one another, reducing the amount of interaction between them.

In the KidPad studies described in this paper we began by investigating a shared view that required social protocol to control navigation, but it was shown to be very frustrating at times for the children. Sometimes one child would begin drawing a picture, and his or her partner would navigate away, and they might not be able to get back to the first users drawing. So in order to study the other significant SDG issues without confounding them with navigation, we opted to not allow navigation at all, and have left this important issue to be investigated in future studies.

Other potential disadvantages of the SDG approach are as follows:

- *Completing tasks might take more time*, because it is no longer possible for a strong willed user to direct the collaboration by controlling the input device.
- *Users may actually collaborate less*. Because they can do work in parallel, they may set about completing their own tasks and never communicate with the other users.

- *SDG applications must squeeze functionality into a very limited screen space*, which may result in reduced functionality compared with similar single user programs.
- *SDG application might be slower* than a single user version, or a traditional groupware system due to increased processing requirements.
- *SDG application might not be portable*, and might exist only for the most popular operating systems. This is because successful SDG implementations today depend on low-level operating system and windowing system interfaces.

In order to build successful SDG applications, these tradeoffs will have to be carefully balanced for each application.

Future directions

We have made a first significant effort at understanding Single Display Groupware in the context of children using local tools as the primary interface mechanism. While we have learned a great deal, much work remains in applying these techniques to other user groups, and with other user interface techniques. In addition to investigating how SDG may apply to more traditional windows-based graphical user interfaces, techniques for managing global navigation are crucial for many applications. In addition, it is important to perform studies analyzing long-term use. Will users learn new forms of collaboration with prolonged exposure to better collaborative tools? Finally, one of the most important areas of research that needs to be investigated further is the area of shared navigation. The research described in this paper has barely brushed the surface of this very rich area. We are continuing to develop KidPad, and are now starting to explore shared navigation approaches.

Recommendations for future systems developers

We were able to build our own architecture for SDG based on local tools. But in order to implement it we were forced to work around a number of small but challenging system limitations from the operating system and the application toolkits. We would like to close this article with a list of suggestions to future computer system developers to consider so that in the future it will be simpler to build collaborative computer applications.

Access to all input devices

It is important for the operating system to provide simple access to all available input devices. Since this work was completed, USB ports have become common which directly support multiple mice. However, in order to access these mice, special-purpose programs must be written as standard input libraries do not support access to the multiple input streams.

No privileged devices

The final step in providing access to input devices and providing a system level representation (cursor) for each input device is to truly make all devices equal in the eyes of the operating system so that any device can interact with any operating system level interface.

Multiple cursors

Currently no operating system provides more than a single cursor. The remote groupware community (e.g., Roseman & Greenberg, 1996) has discussed this significant problem for many years, and it is just as significant for supporting co-present collaboration.

4 ways future system designers can help support SDG:

- Access to all input devices
- No privileged devices
- Enable multiple cursors
- Eliminate global data

Eliminate global data

We need to stop assuming that there will only be a single user or a single input device. All places that store user/device information in global variables need to be eliminated. Instead, this information needs to be placed in data structures that are accessed on a per-device or per-user basis.

Conclusion

This paper describes a model for co-present collaboration that we call Single Display Groupware. Several research groups have recently developed forms of SDG. We have described a framework that may help in understanding common problems, and to suggest ways that technology developers can incorporate low-level support for SDG into their systems.

The usability studies conducted to date, both by ourselves and by others, have indicated that existing technologies have a number of shortcomings when used for co-present collaboration. It appears that SDG technology enables new interaction modalities and can reduce some of the shortcomings observed with existing technology. It also may create new interaction problems. To better understand the overall impact that SDG can have, and to better design SDG applications, longer-term naturalistic studies are needed, and we hope that many people will continue to develop and evaluate SDG technologies and systems.

Acknowledgements

We would like to thank Angela Boltman and the Hawthorne elementary school students in Albuquerque, NM for making the user study of KidPad possible. In addition, we appreciate the CHIKids at CHI 96 and CHI 97 who evaluated early versions of KidPad. Finally, this work could not have been done if it weren't for the other members of the Pad++ team, especially Jim Hollan and Jon Meyer. This work, and Pad++ in general has been largely funded by DARPA to whom we are grateful. We also appreciate Elaine Raybourn's comments on this paper. Finally, we want to thank our colleagues at the University of Maryland who have worked with us in our continued explorations of Single Display Groupware.

References

1. Bederson, B. B., Hollan, J. D., Druin, A., Stewart, J., Rogers, D., & Proft, D. (1996). Local tools: An alternative to tool palettes. User Interface and Software Technology: UIST 96 (pp. 169-170). ACM Press.

2. Benford, S., Bederson, B. B., Åkesson, K.-P., Bayon, V., Druin, A., Hansson, P., Hourcade, J. P., Ingram, R., Neale, H., O'Malley, C., Simsarian, K. T., Stanton, D., Sundblad, Y., & Taxén, G. (submitted). Designing Storytelling Technologies to Encourage Collaboration Between Young Children. Human Factors in Computing Systems: CHI 2000 ACM Press.

3. Berkovitz, J. (1994). Graphical interfaces for young children in a software-based mathematics curriculum. Conference Companion of Human Factors in Computing Systems: CHI 94 (pp. 247-248). ACM Press.

4. Bier, E. A., & Freeman, S. (1991). MMM: A user interface architecture for shared editors on a single screen. User Interface and Software Technology: UIST 91 (pp. 79-86). ACM Press.

5. Bricker, L. J. (1998). Collaboratively Controlled Objects in Support of Collaboration. Unpublished doctoral dissertation, University of Washington, Seattle, Washington.

6. Buxton, W. (1994). The three mirrors of interaction: A holistic approach to user interfaces. L. MacDonald, & J. Vince (eds.), Interacting with virtual environments . New York: Wiley.

7. Druin, A. (Ed.), (1999). The design of children's technology. San Francisco, CA: Morgan Kaufmann.

8. Druin, A., & Solomon, C. (1996). Designing multimedia environments for children: Computers, creativity, and kids. New York: Wiley.

9. Druin, A., Stewart, J., Proft, D., Bederson, B. B., & Hollan, J. D. (1997). KidPad: A design collaboration between children, technologists, and educators. Human Factors in Computing Systems: CHI 97 (pp. 463-470). ACM Press.

10. Greenberg, S., & Boyle, M. (1998). Moving between personal devices and public displays. Calgary, Canada: Department of Computer Science, University of Calgary.

11. Grossmann, H. (1995). Classroom behavior management in a diverse society. Mountain View, CA: Mayfield Publishing Company.

12. Hall, E. (1966). The hidden dimension. Anchor.

13. Heller, S. (1998). The meaning of children in culture becomes a focal point for scholars. The Chronicle of Higher Education, A14-A16.

14. Inkpen, K. M. (1997). Adapting the human-computer interface to support collaborative learning environments for children. Unpublished doctoral dissertation, The University of British Columbia, British Columbia, Canada.

15. Inkpen, K. M., Booth, K. S., Klawe, M., & McGrenere, J. (1997). The effect of turn-taking protocols on children's learning in mouse-driven collaborative environments. Graphics Interface: GI 97 (pp. 138-145). Canadian Information Processing Society.

16. Ishii, H., Kobayashi, M., & Arita, K. (1994). Iterative design of seamless collaboration media. Communications of the ACM, 37(8), 83-97.

17. Krueger, M. (1991). Artificial Reality II. Addison-Wesley.

18. Mateas, M., Salvador, T., Scholtz, J., & Sorensen, D. (1996). Engineering ethnography in the home. Extended Abstracts of Human Factors in Computing Systems: CHI 96 (pp. 283-284). ACM Press.

19. Milligan, C., & Murdock, M. (1996). Testing with kids teens at IOMEGA. Interactions, 3(5), 51-57.

20. Moran, T. P., Chiu, P., & van Melle, W. (1997). Pen-based interaction techniques for organizing material on an electronic whiteboard. User Interface and Software Technology: UIST 97 (pp. 45-54). ACM Press.

21. Myers, B. A., McDaniel, R. G., Miller, R. C., Ferrency, A. S., Faulring, A., Kyle, B. D., Mickish, A., Klimovitski, A., & Doane, P. (1997). The amulet environment: New models for effective user interface software development. IEEE Transactions on Software Engineering, 23(6), 347-365.

22. Myers, B. A., Stiel, H., & Gargiulo, R. (1998). Collaboration using multiple PDAs connected to a PC. Computer Supported Collaborative Work: CSCW 98 (pp. 285-294). ACM Press.

23. Olson, M. H. (Ed.), (1989). Technological support for work group collaboration. Hillsdale, NJ: Lawrence Erlbaum Associates.

24. Papert, S. (1996). The connected family: Bridging the digital generation gap. Longstreet Press.

25. Rekimoto, J. (1998). Pick-and-drop: A direct manipulation technique for multiple computer environments. Human Factors in Computing Systems: CHI 98 (pp. 31-39). ACM Press.

26. Roseman, M., & Greenberg, S. (1996). Building real time groupware with GroupKit, a groupware toolkit. ACM Transactions on Computer-Human Interaction, 3(1), 66-106.

27. Schneider, K. G. (1996). Children and information visualization technologies. Interactions, 3(5), 68-73.

28. Shu, L., & Flowers, W. (1992). Groupware experiences in three-dimensional computer-aided design. Computer Supported Collaborative Work: CSCW 92 (pp. 179-186). ACM Press.

29. Smith, R. B., O'Shea, T., O'Malley, C., Scanlon, E., & Taylor, J. (1989). Preliminary experiments with a distributed, multi-media, problem solving environment. First European Conference on Computer Supported Cooperative Work: (pp. 19-34). Slough, UK: Computer Sciences House.

30. Soloway, E. (1996). Editorial. Interactions, 3(5), 1.

31. Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., & Tatar, D. (1987). WYSIWIS Revised: Early experiences with multiuser interfaces. IEEE Transactions on Office Information Systems, 5(2), 147-167.

32. Stewart, J. (1998). Single Display Groupware. Unpublished doctoral dissertation, University of New Mexico, Albuquerque, NM.

33. Stewart, J., Bederson, B. B., & Druin, A. (1999). Single Display Groupware: A Model for Co-Present Collaboration. Human Factors in Computing Systems: CHI 99 (pp. 286-293). ACM Press.

34. Stewart, J., Raybourn, E., Bederson, B. B., & Druin, A. (1998). When two hands are better than one: Enhancing collaboration using single display groupware. Extended Abstracts of Human Factors in Computing Systems: CHI 98 (pp. 287-288). ACM Press.

35. Stone, M. C., Fishkin, K., & Bier, E. A. (1994). The Movable Filter as a User Interface Tool. Human Factors in Computing Systems: CHI 94 (pp. 306-312). ACM Press.

36. Strommen, E. (1994). Children's use of mouse-based interfaces to control virtual travel. Human Factors in Computing Systems: CHI 94 (pp. 405-410). ACM Press.

37. Vered, K. O. (1998). Schooling in the digital domain: Gendered play and work in the classroom context. Extended Abstracts of Human Factors in Computing Systems: CHI 98 (pp. 72-73). ACM Press.