Single Display Groupware Research in the Year 2000

Garth B. D. Shoemaker
EDGE Lab
School of Computing Science
Simon Fraser University
garths@cs.sfu.ca

Abstract

Users of computer systems are often best served by having the fewest constraints imposed on their interaction styles. This allows users to choose whatever interaction style they deem to be the most effective in any particular circumstance. The recent growth in Single Display Groupware research is a sign that researchers believe the one-person one-display computer usage paradigm may be overly restrictive, to the detriment of the user. It is believed that allowing more than one user to synchronously interact with one display may be desirable in certain circumstances. This paper first looks at the major technological breakthroughs in the field of Single Display Groupware. Then, an examination is performed of research outside of the field that is deemed to be highly relevant. Following this, a summary of the major published works in the area of Single Display Groupware is performed. Finally, conclusions as to the next logical steps to be taken in Single Display Groupware research are drawn.

Keywords: Single Display Groupware (SDG), CSCW, CSCL, toolkits, applications, awareness, privacy

1 Introduction

The first computers were very complex devices that could only be operated by specially trained technicians and scientists. It was eventually realized that computers could not only benefit specially trained users, but average people as well. This prompted a big push in the development of user interfaces, which is evidenced by the birth and growth of the research area of human-computer interaction. In these early days, advancement in the field was difficult, there was no interface standard or research base on which to work. Eventually, based mostly on work from Stanford, Xerox, and Apple, a first primitive interaction standard was established. This standard, involving a mouse driven window and icon-based interface was a necessary first step. These early interfaces are examples of what could be considered the first generation of computer interfaces for average users. Unfortunately, this interface standard has become so well accepted that alternate methods of interaction are often seen as being unnecessary, or worse, undesirable, before any attempt is made to determine validity.

One assumption of the first generation windowed systems was that the system offered a personal view to a single user. Even though the computer was often networked and provided access to remote resources, the user had complete control over the contents of the interface. With the popularization of computers in the workplace, however, it was seen that computer mediated collaboration between users would be desirable. First generation windowed interfaces gave way to what could be considered second generation windowed interfaces. These interfaces offer collaborative tools such as video chat, distributed collaborative editors, and other tools that take input from a variety of distributed users. What differentiates the two generations of interfaces is that in the second generation, window components can be dependent on input from remote users. Even today, these second generation interface tools are rarely offered as default in windowing system software, and are only available through third party application software.

The development of second generation windowing systems was the first step in breaking the ingrained notion that computers should only be viewed and controlled by individual users. The second and final step in breaking this notion will be with the understanding that individual computers can be used by groups of co-located users in collaborative tasks. This, the third generation of windowed interfaces, will free computers to be used in whatever collaborative environments are most appropriate, whether it be a single user with a single computer, multiple users with multiple computers, or multiple-users with a single computer.

This paper investigates the current state of research in the development of the third generation of windowed interfaces. This research, often referred to as Single Display Groupware [27], has delved into both the technical aspects and the sociological aspects of user interactions. As such, toolkits that have been developed to support Single Display Groupware, as well as field research pertaining to the topic will be discussed. Furthermore, relevant research results in other areas, such as second generation distributed interfaces and first generation single user interfaces will be discussed. Finally, open questions for future research will be proposed.

2 Technical Developments

2.1 Toolkits

Most of the technical development in the area of Single Display Groupware has been concerned with overcoming the limitations of computer input systems. It is normally difficult for a computer to receive input from more than one keyboard or mouse at a time. As these are the most commonly used input devices, as defined by the first generation windowing system standard, it is difficult for the computer to gather input from multiple co-located users.

2.1.1 MMM

Perhaps the earliest developed system allowing simultaneous input into one computer by several co-located users is the MMM architecture [2], developed by Bier and Freeman for investigation into the use of shared editors. The architecture provides a windowing system which includes multi-user components. These multi-user aware components include home areas for displaying user-specific information, multi-user menus (essentially lists of buttons), and generic editor components that can be customized for specific editing tasks. The components offer a wide range of support for shared-input, such as simultaneous editing of text strings, and simultaneous reshaping of windows. Scenarios in which Bier and Freeman see their architecture being useful include, among others, the debugging of code on a single workstation by multiple programmers, the use of a large blackboard-like display by users with styli, and the sharing of a handheld computer. Unfortunately, little work was done with MMM after the initial development and early research. The architecture is no longer supported.

Among the issues raised by the MMM project, the most obvious concerns the use of onscreen real-estate and the resulting potential user conflicts. It is evident that indications of individual state, for example toolbars indicating selection mode, have to be visible to each user. As each user may be in a different state, this information must be repeated for each user. Problems may arise if a user confuses another user's state indicator with his own. In addition, the need for multiple state indicators could lead to a great amount of screen clutter. This issue of screen real-estate use is a question that has remained open.

2.1.2 COLT

A more recent toolkit designed with the goal of simplifying the development of co-present collaborative applications is Colt [4]. The Colt software system consists of several parts: a software toolkit for creating Cooperatively Controlled Objects (CCOs), and several additional tools for visualization and analysis. The design methodology behind Colt stresses the creation of CCOs, objects whose state can simultaneously be changed by input from two or more users. The use of CCOs can lead to the creation of required cooperative interaction conditions, in which coordinated input from several users is required in order to reach a desired system state. The Colt system appears to be a useful new addition to the arsenal of SDG toolkits. The inclusion of CCOs in the design methodology behind the system is particularly intriguing. One drawback that should be noted is that the system currently only supports simultaneous input using one of two specific proprietary hardware interfaces.

2.1.3 MID

A toolkit currently under development is MID (Multiple Input Device) [13], a Java API for the creation of multiple input device applications. Although currently limited in terms of supported platforms, MID could potentially offer a universal solution for supporting arbitrary multiple input devices across all platforms supporting the Java standard.

2.1.4 Pebbles Toolkit

The Pebbles project at Carnegie Mellon is based upon the use of the Pebbles Toolkit [19], an extension to the Amulet toolkit [18], that allows for the simultaneous interaction of multiple PDAs with a single computer. The toolkit supports both 3COM PalmPilot-based PDAs and PDAs that run the WindowsCE operating system. The toolkit has built-in support for the creation of independent user input streams, as well as independent undo streams.

2.2 Applications

There are several other recent technological developments in Single Display Groupware. Although they are not toolkits with which to build applications, and are therefore less useful for the development of new applications, it is nevertheless important to know of their existence.

2.2.1 Pebbles Applications

The application side of the Pebbles project, already mentioned in 2.1.4, involves the creation of a variety of tools that allow users to interact with a PC using a PDA [20]. Single Display Groupware applications of note include remote cursor control and keyboard control applications, all of which can be run on a PalmPilot to control software on a shared display.

The concept of employing small personal displays as input devices in an SDG setting is an intriguing one, and could offer a possible solution to the dilemma of how to use screen real-estate for individual user state indicators. Although the early Pebbles applications haven't specifically dealt with the separation of private information from public information, this thread of investigation has been looked into by other researchers [8, 22].

2.2.2 Augmented Surfaces

Another project that involves an original use of existing technology is the Augmented Surfaces project [23]. The augmented environment described in the paper is essentially a large single display that is created by "splicing" together the private displays of individual users who are co-present. A physically seamless and continuous space is created by filling in the gaps between private displays with projected desktop surfaces. For example, when a group of users places their laptops on a table, the laptops communicate with each other and create a shared desktop. A projector on the ceiling of the room is used to project images on the table, filling in the gaps between the private laptop displays. A unified single display is created.

This ability to spontaneously create shared displays is potentially very important, as it may remove the difficulties in synchronizing data between private machines and public machines. All machines will be both private and public, although not simultaneously. As well, screen real-estate problems could potentially become non-issues. Displays large enough to accommodate users' needs could be created on demand, limited only by the available flat surfaces on which to project an image.

2.2.3 Pick-And-Drop

Rekimoto has developed a system promoting novel interaction styles with a shared whiteboard [22]. The development of the system was prompted by the realization that interaction styles suited to single-user systems do not necessarily translate well into a multi-user whiteboard environment. His "pick-and-drop" metaphor promotes the use of a personal tool palette, usually in the form of a PDA, which is similar in functionality to the paint palette of a painter. Each user carries a PDA and can use it to access personal data, or create new data. This provides a convenient separation of personal and public workspaces, in a manner similar to that of the system developed by Greenberg et al. [8], discussed in Section 5.3. What differentiates Rekimoto's system from the other systems mentioned is the stress on real-time creation of data on the PDA, and the resulting pick-and-drop actions to transfer the data to the shared screen.

2.2.4 MIDDesktop

MIDDesktop is a desktop navigation environment built with the MID toolkit that allows any JApplet to be run as an SDG application with independent cursors for each mouse plugged into the computer [26]. It is important because most SDG research involves the time consuming development of custom software. The availability of special toolkits reduces the burden somewhat, but does not eliminate it. MIDDesktop removes the need for development of custom applications, making it possible for research to be performed with existing single user software. MIDDesktop is not suitable for all situations, however. The JApplets running in the MIDDesktop environment are "tricked" into thinking there is one user, when there are really multiple users. This means that complicated coordinated tasks, such as two users simultaneously manipulating the same widget, are impossible. Situations in which the JApplet knows which user is performing what task are also not possible.

3 Distributed Groupware Research

As the study of Single Display Groupware is a relatively new phenomenon, there is a somewhat limited base of research on which to build. It is thus useful to look at other fields in order to find related and relevant ideas that can be adapted to SDG. An obvious area of study in which to look is that of distributed groupware, the second generation of graphical user interfaces discussed in the introduction.

3.1 Awareness

One issue that is prevalent in the area of distributed groupware and relevant to Single Display Groupware research is the question of awareness. In distributed groupware, awareness is important because remotely located users have no direct contact with each other. They must be able to deduce the actions and intentions of remote users through awareness widgets displayed on the screen, or through other media such as audio or haptic feedback. Awareness is also an issue in the design of Single Display Groupware systems, although the problems are sometimes different.

3.1.1 Awareness Widgets

There is much distributed groupware awareness research concerning how to best display awareness information to users. Much of this research deals with "awareness widgets," the name for on-screen components that convey awareness information. A good discussion of awareness widgets is offered by Gutwin and Greenberg [10], and Gutwin and Roseman [11]. Among the specific awareness widgets discussed in the papers are telepointers, which indicate mouse positions of remote users, and multi-user scrollbars, which indicate scrolling position of remote users. Also discussed are a variety of different workspace views, which help users understand what parts of the workspace remote users are working in.

3.1.2 Awareness Mechanisms

There has been some research on awareness in distributed groupware that is very relevant to the study of Single Display Groupware. Dourish and Belloti, in their research on awareness mechanisms, have indirectly highlighted the importance of awareness in Single Display Groupware [7]. Their paper describes an analysis of the mechanisms of awareness in a variety of distributed collaborative text editors. One mechanism of collaboration, which is termed “informative,” requires users to actively transmit information to other users of the system. Another mechanism, termed “role restrictive,” gives users awareness information by assigning all users roles that restrict their possible actions in the system. After analyzing several shared text editors, it is concluded that these two mechanisms of awareness are undesirable. It is argued that the informative mechanism requires too much effort from the producer and assumes relevance for the consumer, neither of which is particularly desirable. It is also argued that the role restrictive mechanism is undesirable because users of a system often want great flexibility in their roles and allowed actions, and are often able to form and adhere to their own informal roles. One mechanism that the paper suggests is useful is that of “shared feedback.” In this mechanism awareness information is gathered passively and is displayed to other users as part of the shared workspace. Dourish and Belloti believe that this overcomes the shortcomings of the other two mechanisms in that user input is no longer required to produce awareness information, and awareness information is naturally incorporated into the workspace so that it can be easily ignored or consumed by other users.

The conclusions reached in the paper are very significant. Even though the paper deals with distributed awareness, it can be seen that many of the problems discussed are easily conquered by employing a co-located single display system. For example, the “shared feedback” mechanism is unavoidable when using a shared display. However, a new set of problems, including awareness overload, arises when designing systems for use with shared displays. This will be discussed in Section 5.2.

3.1.3 Awareness Tradeoffs

Research performed by Gutwin et al. has also served to highlight the relevance of distributed groupware research to Single Display Groupware [10, 9, 11]. In one particularly relevant paper, Gutwin outlines the tradeoffs required when designing an interaction system for a shared workspace [9]. He surmises that the interaction capabilities desired by individuals often conflict with the desired capabilities of the group. The stated tradeoffs include the power of individual user interactions versus shared awareness information for the group. An example tradeoff given in the paper concerns workspace navigation, or whether individual users are freely able to explore the workspace, independent of other users. Allowing this activity favours individual user control, while forcing users to work in the same space favours group awareness.

It can be seen that Single Display Groupware leans towards the group awareness side of the tradeoff. Indeed, by the very nature of Single Display Groupware, workspace awareness is almost guaranteed. The limitations in individual user control resulting from this imbalance is an aspect of Single Display Groupware that must be investigated. Ideally, a method would be found that maintains the heightened group awareness while increasing individual user control.

4 Single User Interface Research

Although single user interfaces have less in common with Single Display Groupware than other multi-user interfaces, there is still much applicable research in the field.

4.1 Local Tools

The idea of Local Tools [1], as developed for use in the Pad++ architecture, may be particularly relevant to Single Display Groupware research. It can easily be understood that screen real-estate management is a serious problem when using a shared display. The problem arises directly from the fact that there are multiple people who may require the use of multiple tool bars and multiple individual windows. Providing individual toolbars for each user results in screen clutter, while having users share a toolbar results in confusion. The use of Local Tools circumvents this problem by discarding the use of toolbars altogether. Instead, tools are scattered about the workspace. These tools each have a unique function, such as paintbrush or eraser, and can be picked up, used, and dropped. The concept was originally developed to mirror way we work at a real desk where pencils, pens, and erasers are scattered about the work surface. A beneficial side effect, though, is that there is no need for private toolbars or extra feedback mechanisms. All the feedback that is needed is provided by looking at what tool is currently being “held.” This is ideal for use on a shared display, where it may be beneficial to limit the private information being displayed. Local Tools are not without limitations, however. First, as discussed in the paper, some users had difficulties performing the operations necessary to pick-up and drop tools. Also, it can be assumed that to accomodate multiple users, it may be necessary to have multiple copies of the same tool. This may result in as much clutter as if there were toolbars.

4.2 Transparency

Another single user interface display technique that may be useful in Single Display Groupware is the use of transparency. Cox et al. suggest using a transparent overview layer to provide a single user with a view of the entire workspace, while the main opaque layer provides a view of the portion of the workspace being manipulated [6]. The results presented in the paper, although limited, provide evidence that users can easily shift attention between the transparent overview layer and the main opaque layer. It appears that users can completely filter out the information on the overview layer when it is not needed, or they can use the overview layer information as a reference while simultaneously manipulating in the main layer. This paper is relevant to Single Display Groupware because of the result that transparent overviews can be selectively ignored. In a Single Display Groupware environment, transparency could be used for the display of private information relevant only to a single user. A menu bar or tool bar, for example, could be drawn transparently, so that it could selectively be seen, or ignored.

Another earlier and more in-depth study of transparent layering has been performed with the approach of studying interference resulting from shared attention between different layers [12]. It is reasonable to assume that the more opaque a transparent overview layer is, the more difficult it will be to ignore. Conversely, the less opaque it is, the more difficult it will be to focus attention on it. This would be significant in the design of a Single Display Groupware system because presumably a balance would need to be found where some users could look "through" the overview layer, while other users could look "at" the overview layer. The paper outlines results from an empirical study measuring interference between different layers at different opacity levels. Harrison at al. also provide anecdotal evidence from observing users using a real-world application with transparent dialog boxes.

Although using transparency to provide workspace navigational awareness, as in the Cox paper, may be useful, a more promising use for transparency is in the design of tools. There has already been research in the area of transparent tools, although it was not prompted by the concerns discussed herein. The Toolglass widget interface tools [3], for example, offer transparent tools that are somewhat reminiscent of Local Tools. Originally meant to be positioned with one hand, while the other hand controls operations performed through the transparent portions of the tools, Toolglass widgets offer several potential advantages over standard tools. First of all, similar to Local Tools, there is no dedicated screen real estate. Toolglass widgets are carried on sheets that can be freely moved about and resized. While they resemble toolbars in some ways, they can easily be resized by removing or adding tools, making them more flexible. Also, the Toolglass widget sheets are transparent, so even if a sheet is lying on top of an area, the information in that area is not completely occluded. Using Toolglass tools instead of normal toolbars and tools, any user at the display has a relatively unobstructed view of any information being displayed on the screen. Combining the use of Toolglass widgets with the use of transparent overview layers could potentially offer any user an unobstructed view of the entire workspace at any time, although it is possible that the use of many different transparent objects would overly complicate the workspace.

5 Single Display Groupware Research

The previous sections described work relevant to, but not directly related to, Single Display Groupware research. This section outlines what are considered to be some areas that need to be focussed on by Single Display Groupware research. Existing research in each of these areas is discussed.

5.1 Foundational Research

Much of the research that has been carried out in the area of Single Display Groupware has focussed on specific users operating in a specific domain at a specific task. It is important for this work to be carried out, however it is perhaps more important that a fundamental and general understanding of user interactions in a Single Display Groupware environment be achieved. By this it is meant that the fundamental effects of having a single display shared by multiple users, regardless of user type, task, or other factor, must be understood. These effects don't all necessarily fall under the same category. There may be any number of effects on inter-user awareness, user attitudes towards the computer, formation of mental models, or other social or cognitive factors. There has been no foundational research performed on these fundamental issues, most likely because the effects are so difficult to isolate. It is necessary that this be done, however, as the results of all other research have basis in these effects.

5.2 Awareness

As opposed to distributed Groupware systems, Single Display Groupware systems implicitly contain a large amount of awareness information. Since each user is looking at the same screen, the full array of required personal information for each user must be displayed. Thus, each component on screen automatically acts as an "awareness widget," meaning it serves to provide state information to users other than the one it directly affects. This is different than distributed systems, where screen components must be explicitly programmed to provide awareness information to remote users.

Although, in Single Display Groupware systems, all users have direct physical and verbal contact with one another, the use of awareness widgets still play a large role. It would be wasteful and impractical to force each user to explicitly communicate to the other users personal state information and intended courses of action. Awareness widgets embedded in the shared workspace can perform much more efficiently.

There is, however, a negative side to having all user components serving as awareness widgets. First of all, there is the fact that having so many shared widgets can cause screen clutter. This effect, known as the screen real-estate problem, not only makes it difficult to fit everything on the screen, but can potentially cause user confusion through awareness overload. If information is constantly being provided to users from a large number of awareness widgets, it may become impossible for a user to mentally process the large volume of information, resulting in important information being missed. Not only can awareness overload result from the sheer volume of information being displayed, but it can also be caused by the presentation of seemingly contradictory information. For example, if each user has a mouse cursor on the screen, there is not necessarily any cue to help a user identify which cursor belongs to whom. The distinction between awareness information and personal state information becomes unclear to the user.

The conclusion that can be reached, interestingly enough, is that Single Display Groupware systems by default have a maximum of awareness information, and the problem is to remove enough of it to optimize group awareness while still providing enough information for individual users to have comprehension of personal state information. This is in contrast to distributed groupware systems where typically group awareness is lacking and awareness information must be added.

The study of awareness in Single Display Groupware is another area in which little research has been performed. A few studies, most notably by Bier et al. [3], Bricker et al. [5], and Inkpen et al. [16], touch on aspects related to the topic of user awareness, but none focus directly on the topic.

5.3 Personal vs. Private Information

Much like the Pebbles project discussed in 2.2.1, Greenberg et al. propose the use of small handheld computers to carry users' personal information [8]. These small personal computers can be easily transported, and can connect to any computer to synchronize data. Thus, whenever a group of users gathers around a shared display, their personal handheld computers can update the shared display with whatever personal information is required from each user. Dealing with information access is already a slight departure from the similar Pebbles project, however, Greenberg goes even further and examines the potential uses of separating public and private information. For example, there may be certain information on a user's personal computer that may not be ready for publicizing to co-workers. That user can choose when to make it public, or may choose to make it public in an anonymous manner, or may choose never to make it public. The information that is public could be displayed on the shared display, while the private information could be made available on the personal computer, accessible only to the appropriate user. Thus, having a personal portable computer in conjunction with shared displays not only allows for easy access to a user's work, but also allows for control over who gets to see that work.

Current work underway by Shoemaker is investigating the feasability and usefulness of providing private information in a true SDG environment [25]. The systems described by Myers et al. [20] and Greenberg et al. [8] provide privacy, but do so by using separate private display, thereby circumventing the shared display. Shoemaker is attempting to develop a system that can show private information on a public display, but make the information selectively visible to certain users, and not to others.

5.4 Education

The field of education is an area that should be closely examined in the context of Single Display Groupware. Not only do modern grade-school curriculum specifications emphasize work in pairs and teams, but the economic situation in many schools rules out the possibility of providing each child with a personal computer. Thus, current situations make it important that Single Display Groupware for children in a learning environment be studied.

5.4.1 Mathematical Coordination

Mathematics is a specific area of education where it is felt that problem related discussion between students can benefit the learning process. Bricker proposes four possible collaborative mathematical exercises that can be used to encourage cooperation and discussion of ideas [5]. The exercises, which have been implemented as SDG computer applications, force collaboration between users by requiring that a part of a problem's solution be given by each user. For example, one exercise, a three-user exercise, requires the users to match two colour samples by changing the RGB values in one of them. Each user can only change one value, R, G, or B. It is hoped that this will encourage problem-related discussion, and will promote higher-order thinking.

5.4.2 Inter-User Coordination

When considering the use of SDG groupware, especially in an educational environment, it is necessary to consider the social implications of placing multiple users at a shared display. Inkpen et al. have investigated the changes in behaviour, as well as performance, that occur when providing different access protocols to pairs of school-aged children playing a puzzle-solving computer game [16, 14]. In the study, the children each had a mouse, but only one child could control the game at a time. In the "take" access protocol, the child without control could preemptively acquire control. In the "give" protocol, the child with control had to voluntarily relinquish control. As Stewart points out, the positive learning improvements evident in Inkpen's study are important because they signify that SDG could benefit tasks in which both users are not expected to work simultaneously [27].

Building on Inkpen's past educational work, a recent study by Inkpen et al. has shown that not only can performance be improved in a multi-input shared display setting, but activity levels of users can also be increased in the same situation [15]. In the study, pairs of grade-school students were asked to solve a pattern matching game in either a paper-based condition, a shared one-mouse computer condition, or a two-mouse two-cursor computer condition. Preliminary results showed that when each user had a mouse and a cursor activity levels were higher, and off-task behaviour was less frequent. More indepth analysis of the data, performed by Scott et al., showed that the children's interactions were more natural (closer to that of the paper condition) in the two-mouse condition than they were in the one-mouse condition [24].

5.5 Business

The business arena presents possibly the most important, and lucrative, market for Single Display Groupware applications. Prior to the computer age, professional designers, writers, and creators in all areas were accustomed to co-located group collaborations when working on a project. The computer age has stifled this process, isolating workers in cubicles. The introduction of well-designed Single Display Groupware systems into the corporate world will result in new workplace dynamics, where workers are free to interact with each other more naturally, rather than being hobbled by the location, accessibility, and abilities of their terminals.

5.5.1 Electronic Whiteboards

One of the technologies that most interests the corporate sector is that of interactive whiteboards. Normal whiteboards are a common tool in business meetings and brainstorming sessions, and the possible advantages to having an electronic version are intriguing.

One of the earliest attempts at introducing Single Display Groupware technology to the corporate community was with the development of Tivoli [21], a software application meant to run on large electronic whiteboards in informal meetings. LiveBoard, the application that was eventually delivered to customers was never well accepted, and really could not be considered a commercial success. The problem was that instead of taking a set of capabilities already available to users and expanding them, LiveBoard restricted the existing set of capabilities, and offered some extra capabilities in exchange. For example, the commercial version of LiveBoard did not seamlessly support simultaneous use of the board by multiple users. This is functionality that is expected by users of a whiteboard. If it is not present, the users will not be entirely happy, no matter what extended functionality is available.

A counter-example to the problems apparent in the Tivoli product is ClearBoard [17], another whiteboard type application, originally developed for distributed collaboration. One of the foundations of the ClearBoard design is that it should act exactly like a real whiteboard. Possibly as a result of this design methodology, ClearBoard can be used with multiple people at either end of the network, even though this was not explicit in the design. This is a perfect example of how an ideal system allows for flexible interaction styles, whether distributed or co-located.

5.5.2 Electronic Tabletops

Another technology that has great potential in the business world is that of electronic tabletop displays. Physical tabletops have always been a key part of business environments. They support natural and rich group interactions, and are particularly well suited for supporting the use of tangible media. In other words, if you put something on a table, it doesn't fall off.

In the context of electronic tabletops as Single Display Groupware devices, one scenario described by Underkoffler and Ishii is particularly relevant [28]. The discussion centers around *Urp*, an electronic tabletop system that combines the use of tangible media and luminous information to provide an environment in which urban planning and design investigations can be carried out. The scenario described in the paper deals with two urban planners gathered around the table, changing the positions of buildings (tangible media) to investigate effects on shadows and reflection angles (luminous information). The two planners are able to discuss and implement changes on the shared workspace, and receive shared feedback from the workspace. This ability to easily interact with the workspace as well as with other planners is possibly the reason why "most of the professional visitors said they would voluntarily begin using a system like *Urp* immediately if it were available."

6 Open Questions

One important issue surrounding the field of Single Display Groupware concerns the benefits that may be recognized by supporting co-located collaborative activities. It is understood that there can be benefits in this type of interaction, however what the benefits are, and why they exist, are poorly understood. It is important that there be research focussing on the fine grained interactions between users in different interaction scenarios, so as to determine the important aspects of communication that result in a comfortable, collaborative experience. Not only would this benefit the understanding of Single Display Groupware activities, but it could also benefit distributed application development. An awareness of the important aspects of inter-user communication could lead to better distributed applications that would help make users feel "closer" to each other.

Another important issue that has recently arisen in the field is the handling of private versus public information. Single Display Groupware deals with shared displays, which are inherently public spaces. This is contrary to all our past experience in computer interaction. Whereas in the past information was by default private, the reverse is now true. The question of what to do with private information: how to display it, how to access it, and how to make it public, is open.

It is necessary that this be further explored.

Closely connected with these past two questions is the issue of user awareness. By studying the specifics of user communication it can hopefully be determined what components of awareness are needed in a collaborative setting. Also, the issue of private versus public information is closely related to awareness. What information is not needed to promote group awareness can be retained as private information through some mechanism, whereas some information should be made public to promote group awareness. The means by which this is done, by the system and by the users, needs to be addressed.

In terms of direct interactions styles, it will be useful to further investigate what physical interaction styles are most useful. It is fairly evident that mice are not ideal for all situations. This paper has discussed the use of stylus type devices for true direct manipulation, as well as the novel use of PalmPilot type PDAs for other types of manipulation. Whether these are truly useful interaction methods is yet to be seen.

Another important issue that has already been mentioned is the problem of screen real-estate. Multiple users surrounding a single display will quickly run out of screen space, and it is possible that screen components meant to inform one user will confuse another user. There is no one solution to this problem. Some possible solutions involve the use of semi-transparent interface components, large-screen displays, local tools, and augmented surfaces.

While all these issues discussed so far are important, the most important issue to be faced is how to have users accept the applications we develop for them. The ultimate measure of how useful a technology is, is to observe users at work or play. If the users choose the technology over the alternatives, then it is a success. If not, then it is a failure, no matter how ingenious it may have been in design. Towards this end of pleasing the users, it is important that researchers keep close ties with the corporate world, as well as educational institutes for both children and adults. The final open question to be answered is "what do the users want?"

References

[1] B. B. Bederson, J. D. Hollan, A. Druin, J. Stewart, D. Rogers, and D. Proft. Local tools: An alternative to tool palettes. In *Proceedings of UIST '96*, pages 169–170. ACM Press, 1996.

[2] E. A. Bier and S. Freeman. MMM: A user interface architecture for shared editors on a single screen. In *Proceedings of UIST '91*, pages 79–86. ACM Press, 1991.

[3] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: The see-through interface. In *Proceedings of SIGGRAPH '93*, pages 73–80. ACM Press, 1993.

[4] L. J. Bricker, M. J. Bennett, E. Fujioka, and S. L. Tanimoto. Colt: A system for developing software that supports synchronous collaborative activities. In *EdMedia*, 1999.

[5] L. J. Bricker, S. L. Tanimoto, A. I. Rothernberg, D. C. Hutama, and T. H. Wong. Multiplayer activities that develop mathematical coordination. In *CSCL '95*, pages 32–39, 1995.

[6] D. A. Cox, J. S. Chugh, C. Gutwin, and S. Greenberg. The usability of transparent overview layers. In *CHI '98*, pages 301–302. ACM Press, 1998.

[7] P. Dourish and V. Belloti. Awareness and coordination in shared workspaces. In *Proceedings of CSCW '92*, pages 107–114. ACM Press, 1992.

[8] S. Greenberg, M. Boyle, and J. LaBerge. PDAs and shared public displays: Making personal information public, and public information personal. *Personal Technologies*, 3(1), 1999.

[9] C. Gutwin and S. Greenberg. Design for individuals, design for groups: Tradeoffs between power and workspace awareness. In *Proceedings of CSCW '98*, pages 207–216. ACM Press, 1998.

[10] C. Gutwin and S. Greenberg. Effects of awareness support on groupware usability. In *Proceedings of CHI '98*, pages 511–518. ACM Press, 1998.

[11] C. Gutwin and M. Roseman. A usability study of awareness widgets in a shared workspace groupware system. In *Proceedings of CSCW '96*, pages 258–267. ACM Press, 1996.

[12] B. L. Harrison, H. Ishii, K. J. Vicente, and W. A. S. Buxton. Transparent layered user interfaces: An evaluation of a display design to enhance focused and divided attention. In *Proceedings of CHI '95*, pages 317–324. ACM Press, 1995.

[13] J.-P. Hourcade and B. B. Bederson. Architecture and implementation of a Java package for multiple input devices (MID). Technical Report CS-TR-4018, University of Maryland, May 1999.

[14] K. M. Inkpen, K. S. Booth, S. D. Gribble, and M. Klawe. Give and take: Children collaborating on one computer. In *Proceedings of CHI '95*, pages 258–259. ACM press, 1995.

[15] K. M. Inkpen, W. Ho-Ching, O. Kuederle, S. D. Scott, and G. B. D. Shoemaker. "this is fun. we're all best friends and we're all playing!": Supporting children's synchronous collaboration. In *Proceedings of CSCL '99*, 1999.

[16] K. M. Inkpen, J. McGrenere, K. S. Booth, and M. Klawe. The effect of turn-taking protocols on children's learning in mouse-driven collaborative environments. In *Proceedings of GI '97*, pages 138–145, 1997.

[17] H. Ishii and M. Kobayashi. Clearboard: A seamless medium for shared drawing and conversation with eye contact. In *Proceedings of CHI '92*, pages 525–532. ACM Press, 1992.

[18] B. A. Myers. The Amulet environment: New models for effective user interface software development. *IEEE Transactions on Software Engineering*, 23(6):347–365, June 1997.

[19] B. A. Myers. An implementation architecture to support single-display groupware. Technical Report CMU-CS-99-139, Carnegie Mellon University, 1999.

[20] B. A. Myers, H. Stiehl, and R. Gargiulo. Collaboration using multiple PDAs connected to a PC. In *Proceedings of CSCW '98*, pages 285–294. ACM Press, 1998.

[21] E. R. Pederson, K. McCall, T. P. Moran, and F. G. Halasz. Tivoli: An electronic whiteboard for informal workgroup meetings. In *Proceedings of INTERCHI '93*, pages 391–398. ACM Press, 1993.

[22] J. Rekimoto. A multiple device approach for supporting whiteboard-based interactions. In *Proceedings of CHI '98*, pages 18–23. ACM Press, 1998.

[23] J. Rekimoto and M. Saitoh. Augmented surfaces: A spatially continuous work space for hybrid computing environments. In *Proceedings of CHI '99*, pages 378–385. ACM Press, 1999.

[24] S. D. Scott, G. B. D. Shoemaker, and K. M. Inkpen. Towards seamless support of natural collaborative interactions. In *Proceedings of GI 2000*, 2000. In Press.

[25] G. B. D. Shoemaker. Supporting private information on public displays. In *Proceedings of CHI 2000*. ACM Press, 2000. In Press.

[26] G. B. D. Shoemaker and K. M. Inkpen. MIDDesktop: An application framework for single display groupware investigations. Technical Report TR 2000-01, School of Computing Science, Simon Fraser University, 2000. ftp://fas.sfu.ca/pub/cs/TR/2000/CMPT2000-01.pdf.

[27] J. Stewart, B. B. Bederson, and A. Druin. Single Display Groupware: A model for co-present collaboration. In *Proceedings of CHI '99*, pages 286–293. ACM Press, 1999.

[28] J. Underkoffler and H. Ishii. Urp: A luminous-tangible workbench for urban planning and design. In *Proceedings of CHI '99*, pages 386–393. ACM Press, 1999.