Avoiding Interference through Translucent Interface Components in Single Display Groupware

Ana Zanella and Saul Greenberg
University of Calgary
Department of Computer Science
Calgary, AB, T2N 1N4 Canada
+1 403 220-7686
azanella@cpsc.ucalgary.ca

ABSTRACT

Our research concerns the design of interface components tailored for *single display groupware* (SDG) where multiple co-located people, each with their own input device, interact over a single shared display. In particular, we are concerned with 'interference' effects, where one person's raising of an interface component (e.g., a menu) can impede another's view and interaction on the shared screen. Our solution uses translucent interface components, where others can see through the obstructing component and continue their work underneath it. Our in-progress evaluation suggests this design lessens interference effects.

Keywords. Single display groupware, translucence.

PROBLEM OVERVIEW AND PROPOSED SOLUTION

Single Display Groupware (SDG) is a class of groupware applications designed to support face-to-face groups who work together around a single computer display. Displays can range from conventional monitors to large rear-projected screens. Each person also has their own input device, configured so they can simultaneously interact with the SDG application [3].

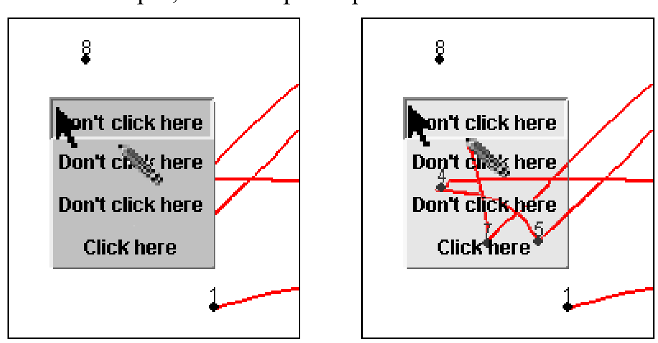
While SDG systems are conceptually simple, there are surprisingly many issues related to their design. Some issues are technical e.g., how operating systems and programming languages support simultaneous use of multiple input devices. Other issues are related to the user interface. Standard interface components (or widgets) are designed to recognize and respond to actions of a single user only. Consequently, they may have ambiguous semantics in a multi-user setting, they may not allow simultaneous interaction over them, or they may not present feedback in a way that is appropriate to the group.

Floating components, interference and translucency

The particular SDG interface issue we are investigating concerns the design and evaluation of floating and transient SDG interface components that, when used by one person, may cause *interference* for other SDG users. Interference occurs when the floating component—pop-up menus, floating palettes, secondary windows, dialog boxes—obstructs another's view and interaction in the area underneath or near the component. For example, one person can unintentionally cause interference when he or she raises a menu over another person's cursor, especially if the other person is in the middle of doing some work. This is illustrated in Figure 1a: the person with the arrow cursor has effectively blocked the person with the pencil cursor from continuing their drawing actions. Similar problems occur when the system raises a secondary window as a side effect of a specific user action e.g., a dialog requesting further information from one user may block others who need to access whatever is underneath.

One solution is to do away with these floating and transient components altogether. For example Druin *et a*l [1] propose the idea of *local tools*, where large, simple tools sitting directly on the work surface would replace traditional floating tool palettes. That is, local tools are guaranteed to appear in the space rather than above it. While reasonable for certain applications (Druin applied these to interfaces for children), we believe it cannot be generalized to all applications. For example, functionally rich applications may have so many tools and options that it would be unreasonable to map each to a simple tool.

We are experimenting with another idea that retains the notion of floating components. First, we make the floating component translucent: all participants not only see what is on the component, but can also see through it to the area underneath. Second, the component responds only to its owner's input; all other participant's actions are directed to



(a) opaque pop-up menu (b) translucent pop-up menu

Figure 1. Two views of our 'connect the dots' SDG game. The *interferer* raises the pop-up menu directly above where the *player* is connecting the line to the next dot.

Zanella, A. and Greenberg, S. (2000) **Avoiding Interference through Translucent Interface Components in Single Display Groupware.** In *Extended Abstracts of the ACM Conference of Human Factors in Computing Systems (CHI'01)*, ACM Press.

Also as *Report 2000-674-26,* Dept Computer Science, University of Calgary, Alberta, Canada, October.
http://www.cpsc.ucalgary.ca/grouplab/papers/index.html

the correct objects underneath it on the work surface. Figure 1b illustrates how this is done on a pop-up menu. The translucency level of the menu is set to make the menu, its items, and the drawing underneath all visible. However, the menu only responds to the person who raised it, while the other person can continue to draw underneath.

To recap: our premise is that translucent floating interface components will mitigate interference effects between SDG users because others can still see through them and continue their work.

USER STUDY

We are currently running a controlled study comparing the usability of translucent and opaque popup menus in SDG. The study is based on pairs of users (all proficient computer users) playing an SDG version of a "connect the dots" game. As will be described below, we measure the degree of interference effects by how well players perform their task and by measures of satisfaction.

We bias the game towards 'worst case' interference between the SDG users. One person, called the *player*, is asked to draw a line connecting all the dots in numeric order: the player draws a line from one point to the next with the left mouse button and then marks the dot as connected by clicking on it with the right mouse button. The other person, called the *interferer*, is asked to impede the player as much as possible by popping up a menu atop of where the player is working: the interferer raises the menu with a right button click and makes a selection with the left button. Because it may be possible for the interferer to block the player indefinitely, the interferer is also asked to chose a menu item as rapidly as possible (the randomly positioned 'click here' item visible in Figure 1).

There are three types of trials in the test.

Solo: only one user plays. That is, the person connects the dots without any interference. This gives a 'best case' performance time for a player to connect the dots.

Opaque menus: both player and interferer play, where the interferer's menus are opaque (as in Figure 1a).

Translucent menus: as above but using translucent menus (as in Figure 1b).

Each game displays 15 randomly positioned dots to be connected. Each pair plays 24 games divided into 8 sets, where each set contains the three different trial types. For each game, we record the time it took for the player to successfully connect all the dots, as well as the number of times an interferer popped up a menu on the top of the other user's cursor (excepting, of course, in the solo condition). After playing all games the participants filled out a post-session questionnaire asking them about their menu preferences and the way they felt the different menu types influenced their tasks.

Preliminary Results

Our study is mostly complete: we have collected data for 30 pairs, but have only partially analyzed the results. While tentative, our results suggest that translucent menus in SDG are promising.

First, user preferences strongly indicate translucent menus over opaque ones in the SDG situation, as illustrated in the table below. 34 of the 60 subjects strongly preferred translucent menus, and 9 more had a weak preference. Only 10 of the 60 liked the opaque menus.

Which type of menu do you prefer (all subjects)?

Opaque ... ***Translucent***

Strong	Weak	Neutral	Weak	Strong
7	3	7	9	34

When asked how the different menus affected their task, almost all players thought that translucent menus made it easier for them to continue their work in spite of interference (28 of the 30 players). On the flip side, almost all interferers thought that translucent menus made it harder for them to interfere with the player (25 of the 30 interferers). The tables below show the specific results.

How do translucent menus help your task?

Players

Easier	Same	Harder
28	2	0

Interferers

Easier	Same	Harder
0	5	25

We are still performing our quantitative analysis of the efficacy of each menu type by comparing how long the player takes to connect the dots across the different trial types. For exploratory purposes, we collapsed the data within each pair into an average time / trial type. In almost all cases, the average time relationships are: solo < translucent < opaque. An exploratory single factor ANOVA suggests these differences are statistically significant ($F=21.38$, $p<<0.01$). We caution that these are preliminary results: we have a few more pairs to run, and we intend to do more refined statistical analyses. Still, we can tentatively conclude from both the quantitative analysis and the questionnaire results that SDG users strongly prefer translucent *vs* opaque menus in SDG, and that translucence interface components mitigates—but does not eliminate—interference in SDG systems.

Acknowledgements. Thanks for Hourcade and Bederson's MID software [2]. We are grateful to Microsoft Research, ASERC, NSERC and Smart Technologies for funding.

REFERENCES

1. Druin, A., Stewart, J., Proft, D., Bederson, B. and Hollan, J. KidPad: A design collaboration between children, technologists, and educators. *Proc ACM CHI'97,* v.1, 463-470.
2. Hourcade, J. and Bederson, B. B. (1999). Architecture and implementation of a Java package for multiple input devices (MID). Report HCIL-99-08, Computer Science Dept, University of Maryland, MD USA.
3. Stewart, J., Bederson, B. and Druin, A. (1999) Single display groupware: A model for co-present collaboration. *Proc. ACM CHI'99*, 286-293.