Windowing System Support for Single Display Groupware

Peter Hutterer and Bruce H. Thomas
Wearable Computer Laboratory
School of Computer and Information Science
University of South Australia
Mawson Lakes SA 5095
+61 8 8302 3669
{peter|thomas}@cs.unisa.edu.au

ABSTRACT

In this demonstration, we present the Multi-Pointer X Server (MPX). MPX is a modification of the X Window System to support multiple independent mouse devices and a fine-grained floor control mechanism that can be applied to any GUI element. The custom-built window manager administers Single Display Groupware (SDG) features such as the floor control and annotation overlays. The windowing system and the window manager is the proper location for software support of SDG. SDG integrated into the windowing system allows for two-handed interaction and the simultaneous multiple user utilisation of legacy and SDG applications.

Categories and Subject Descriptors

H.5.2. [**Information Systems**]: User Interfaces – *Windowing systems*

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces – Computer-supported cooperative work, Synchronous interaction, Collaborative computing

General Terms

Design, Human Factors

Keywords

Single Display Groupware, Windowing System, CSCW, Graphical User Interfaces.

1. INTRODUCTION

Current windowing systems restrict the users to a single pair of keyboard and mouse devices. Research into Single Display Groupware [5] showed that co-located users preferred individual input devices [6], but few SDG toolkits have been adopted outside of research laboratories. Previous SDG toolkits [3, 7] require applications to be built against the toolkit's API, restrict users to using only one application at a time, and often exclude legacy applications. Applications that are used in a common office environment – text processors, spreadsheets, etc. – cannot be used in a multi-user setup.

We believe that the most important feature to quicken the adoption of SDG is the unrestricted support of those applications that are employed everyday. At the same time, future SDG applications require custom APIs to incorporate multi-user facilities. Both types of applications need to co-exist on a single desktop. We hypothesise that the correct level for support of SDG is a combination of the windowing system and the window manager. The windowing system is the only framework that is used by every graphical application. Thus, if SDG is supported natively by the windowing system, any graphical application can be run in a SDG setup where multiple users may use it simultaneously. The window manager is responsible for the placement and visibility of application top-level windows and can thus facilitate additional SDG features.

We have enhanced the current version of the X Window System server to support up to 18 true system cursors. The cursors can operate independently and in multiple applications simultaneously. Our Multi-Pointer Window Manager (MPWM) provides floor control administration policies for legacy applications and per-application annotation overlays. MPX can be used with legacy applications and custom-built SDG applications. Although MPX is based on the X Window System, the model is also applicable to the windowing environments of Microsoft Windows or MacOS X.

In this demonstration, we show the following features of MPX:

- Two-handed interaction with legacy applications such as web browsers, drawing programs or office applications.
- Two-handed interaction with custom-built SDG applications.
- Interaction with multiple applications (both legacy and SDG applications) and multiple users simultaneously.
- Administering a floor control policy from within a SDG application.
- Administering a floor control policy for legacy applications from the window manager.

2. RELATED WORK

Users prefer their own devices instead of sharing input devices [6]. However, SDG applications (i.e. [1] [2]) are mostly limited to only a small set of functionality. SDG toolkits such as [3] [7] require applications to be compiled against a new API. TIDL [4] supports legacy applications but is restricted to applications that employ the Java Swing toolkit. No SDG environment or toolkit currently provides a generic architecture for any graphical application.

3. MPX

A standard X Server merges events from multiple physical input devices into a single cursor. MPX assigns one cursor per configured device. Any graphical application is directly dependent on the windowing system's API and a change in the windowing system is thus reflected in those applications. When multiple mice are introduced on the windowing system level, legacy applications can be used in multi-user contexts. Each cursor in MPX behaves like a system cursor and MPX is thus compatible to most existing single-user applications.

MPX supports an elaborate floor control mechanism that can assign different permissions to any GUI element. This floor control can be very elaborate as each menu, checkbox, text field, button or application can have a separate access list. For example, in a drawing application multiple users may be allowed to draw on the canvas but only one person may press the button to save the current drawing to disk. A floor control policy enforced in the windowing system ensures that only devices that have the necessary access rights deliver events.

MPX does not introduce new APIs. Applications that need to utilize multicursor support can employ the existing X Input Extension (XI) which has been modified to accommodate MPX. Any XI event contains the device ID of the input device and an application can thus identify the device that generated a particular event. This allows for features such as simultaneous drag-and-drop or user dependent states. The XI API is well established and utilised in higher-level toolkits such as GTK. One example for an SDG application that employs the MPX functionality is our multi-pointer window manager (MPWM).

4. MPWM

A window manager is responsible for the window placement, controls visibility of windows and the window decorations. Thus the window manager is the ideal place to integrate SDG administration interfaces, especially for applications that do not actively employ SDG.

The Multi-Pointer Window Manager is designed as a reference implementation to demonstrate how multi-cursor support may be integrated into a window manager developed with the MPX additional functionality. A number of legacy applications can be used with multiple cursors, but they do not control SDG functions provided with MPX. The window manager knows the placement of all application windows and can thus provide floor control support on a per-window basis. Custom-built SDG applications can communicate with the window manager and tell it to leave floor control functionality to the application. MPWM also provides a per-window annotation layer. The annotation layer is tied to the application window. While one device annotates, all other users can access the underlying application. When the application window is moved, the annotation layer moves with the window. Each device has a separate and distinctively coloured layer on the annotation. Annotation works for any application window including legacy applications.

4.1 DEMONSTRATION

In this demonstration, we will show how multiple users can simultaneously interact in a SDG windowing environment. Multiple legacy applications such as web browsers or drawing programs can be operated simultaneously. We will also show how two-handed interaction can be employed for legacy applications. A set of custom-built applications demonstrates how MPX features can be applied in SDG applications using existing and well-established APIs. Finally, we will demonstrate the floor control capabilities of MPX.

5. CONCLUSION

We believe that support for independent input devices has to be integrated into the windowing system and the window manager to facilitate the widespread adoption of Single Display Groupware. Our modifications to the X Window System support multiple true system cursors and a floor control mechanism to assign access rights to every GUI element. In combination with a multi-cursor aware window manager we can execute both legacy and groupware applications simultaneously. The window manager supports two-hand interaction for window operations and provides an annotation layer for each application window. The floor-control is administered by the window manager but can also be controlled by an application.

6. REFERENCES

[1] Bier, E.A. and Freeman, S. MMM: a user interface architecture for shared editors on a single screen. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*. (Hilton Head, South Carolina, United States, 1991). ACM Press 1991, 79-86.

[2] Crowley, T., Milazzo, P., Baker, E., Forsdick, H., and Tomlinson, R. MMConf: an infrastructure for building shared multimedia applications. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*. (Los Angeles, California, United States, 1990). ACM Press 1990, 329-342.

[3] Hourcade, J.P. and Bederson, B.B., *Architecture and Implementation of a Java Package for Multiple Input Devices (MID)*. Technical Report CS-TR-4018. University of Maryland, College Park, MD, 1999.

[4] Hutterer, P., Close, B.S., and Thomas, B.H. TIDL: Mixed Presence Groupware Support for Legacy and Custom Applications. In *Proceedings of the seventh conference on Australasian user interfaces*. (Hobart, Australia, 2006). Australian Computer Society, Inc. 2006, 107-114.

[5] Stewart, J., Bederson, B.B., and Druin, A. Single display groupware: a model for co-present collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. (Pittsburgh, Pennsylvania, United States, 1999). ACM Press 1999, 286-293.

[6] Stewart, J., Raybourn, E.M., Bederson, B., and Druin, A. When two hands are better than one: enhancing collaboration using single display groupware. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*. (Los Angeles, California, United States, 1998). ACM Press 1998, 287-288.

[7] Tse, E. and Greenberg, S. Rapidly prototyping Single Display Groupware through the SDGToolkit. In *Proceedings of the fifth conference on Australasian user interfaces*. (Dunedin, New Zealand, 2004). Australian Computer Society, Inc. 2004, 101-110.