Visual Manipulations for Improved Generalized Presentations

Lior Berry, Kellogg S. Booth and Lyn Bartram

Department of Computer Science
University of British Columbia

Abstract

In many cooperative work sessions a presenter shares a view of her workstation screen with a group of passive viewers, projecting it on a large screen or using bitmap-based sharing protocols. While these schemes offer advantages of simplicity and sharing transparency over custom collaboration-aware tools, all group members will have the exact same view. Data deemed private by the presenter may be exposed and viewers have to watch the presenter's detailed interactions, making it hard to infer her underlying intentions. We present a general system that allows customization of existing single user application views. Simple policy-guided manipulations are applied to the shared windows and image buffers, providing proactive privacy protection with relaxed verbosity and cues to meet presenter and viewers' needs.

Key words: CSCW, Application, Sharing,, Privacy.

1 Introduction

Popular *generalized presentation* scenarios require a single person, the *presenter*, to share a view of her workstation, while others, the *audience*, watch (for instance, classroom and conference presentations, training and demonstration sessions or group editing, where only one person makes changes as others watch).

View sharing is often done by projecting the presenter's desktop onto a public screen or using simple bitmap-based screen sharing protocols like VNC (www.realvnc.com). These collaboration-transparent solutions are equivalent and offer key advantages: (i) any application can be shared, without code changes; (ii) viewers do not need a copy of the application; and (iii) presenter and audience get synchronized views (required for referral transparency).

The latter advantage is also a chief drawback. Strictly identical views fail to address the presenter's privacy and the difficulties faced by passive viewers. The motivation for our work is the need to address these, without giving up much of the aforementioned advantages that still make these simple sharing schemes a favorable collaboration fallback (over collaboration-aware tools or synchronized document replications). Therefore, improvements to these modes are still highly viable for real-world collaboration.

Sharing specific applications rather than the entire desktop partially solves the privacy problem. Harder to control are application-specific objects (e.g. a range of spreadsheet cells or a paragraph that need to be private, yet available for the presenter) or UI widgets with sensitive or embarrassing information, "leaking" in awkward moments (e.g. an error dialog, IM messages, navigation history and "recent files") that require the presenter to stay alert and feel uncomfortable.

Passive viewers, trying to infer underlying intentions and semantics from the presenter's interactions need to control detail levels (verbosity), display clutter and adjust GUI parameters that are tuned for an active user (menu placement, invisible keyboard shortcuts or interaction tempo). They also need visual cues to accurately follow the presenter.

We describe a proactive framework for balancing privacy and awareness concerns. Transparent sharing bitmap-based techniques are augmented with a novel set of manipulations to meet both general and application specific concerns.

2 System Description

Sharing a view is carried out by grabbing the image buffers of shared application windows and cloning them on the public display (similar to [1]). Yet, the replication pipeline introduces an extra step to apply manipulations to the buffers before being "published".

We need to locate the visual representations of application objects, UI widgets or windows for effective manipulations. The system uses an extensile plug-in architecture, in which a base plug-in provides generalized services, (e.g. identifying private menus and dialogs). Application-specific middleware plug-ins ("semantic glue") can extend it, supporting a common API. They determine private object locations, areas for highlighting, application states or windows to clone. A plug-in translates general queries into application specific queries in three layers: OS APIs, Accessibility APIs and Application specific automation APIs.

Identifying private objects or areas for highlighting can be based on specific markings or hints given by the presenter (e.g. mapping a specific object "magic marker" color to privacy, or querying active selection). A complementary approach uses rule-guided searches for elements that may be private (e.g. searching the

Accessibility and window trees or the document text for personal information, network or security settings etc.). The system can then apply different manipulations to these elements (Figures 1and 2).

Imager Buffer Manipulations: *Blur* filters issued on the visual representation of private elements (in the public view) enable the presenter to work normally, while protecting her privacy. They offer varying degrees of awareness for viewers.

Highlighting the "*active context"* of presenter's interactions (e.g. the paragraph containing selection or the table surrounding selected cells) or keeping a key object highlighted while interacting elsewhere, can assist viewers. Highlighting can also be applied to detected document changes or to menu selections.

Spatial manipulations: Sharing only window parts can be useful for mitigating clutter ([1]). We support automatic stripping of space taking UI layers (like toolbars) from the public view. Alternatively, only the window part containing active context or a selected object can be shared, even when presenter resizes or scrolls the window. Other manipulations can move obscuring menus or automatically downsize dialogs and wizards.

Temporal or State manipulations: In some states it is better not to update the public view to maintain privacy (e.g. opening the navigation history, an error message popping or exposing comments) until the presenter exists the state. In other cases we can replace verbose interactions or non-visual interactions with an iconic indication to provide awareness for viewers (replacing a file open dialog with an icon or a menu selection and keyboard shortcuts with a subtitle).

2.1 Related Work

VNC or MS Live-Meeting (www.placeware.com) allow sharing of entire screens or specified applications, but do not manipulate visuals, ignoring privacy and viewer needs. The presenter has to constantly address these.

Surveys of collaboration-aware solutions as well as sophisticated replication based solutions that can be crafted to meet the needs we identified are presented in [2] and [4]. However, these are often bound to specific applications or architectures; require non-trivial synchronizations or assume all parties have a copy of the application.

In [3] a limited set of visual manipulations (like highlighting) is applied to simple applications, but only to support the work of a single user.

Spatial manipulations, based on manual marking of window parts to share, are presented in [1]. They quickly breaks down as windows are resized or scrolled and are completely subsumed by our system.

Commercial screen recording tools (e.g. Camtasia, www.techsmith.com) allow separate editing sessions to add highlighting, annotations or clean up recorded interactions to assist viewers. Our system provides similar functionality but in real-time.

3 Future Work and Conclusions

We intend to look into other automated schemes for extracting privacy leaks and policies to handle them (e.g. "program by example" and machine learning techniques), including the adaptation of the semantic glue for fine grained access control. A formal user study is planned on the next version of the system.

The presented framework disproves to some extent the misconception that bitmap-based application sharing forces strict WYSIWIS ([4]). By using simple, generalizable manipulations it effectively protects the presenter's privacy, allowing her to work comfortably and assists viewers in understanding her intentions.

References

[1] Tan D. S., Meyers B., Czerwinski, M., Wincuts: manipulating arbitrary window regions for more effective use of screen space. In Proceedings of CHI, pages 1525-1528, 2004.

[2] Xia S., Sun D., Sun C., Chen D., and Shen H., Leveraging single-user applications for multi-user collaboration: the coword approach. In Proceedings of CSCW, pages 162-171, 2004.

[3] Olsen, D. R., Hudson, S. E., Verratti T., Heiner J. M., and Phelps M., Implementing interface attachments based on surface representations. In Proceedings of CHI, pages 191–198, 1999.

[4] Begole, J., Rosson, M. B., and Shaffer, C. A., Flexible collaboration transparency: supporting worker independence in replicated application sharing systems. In TOCHI, 6(2), pages 95-132, 1999.

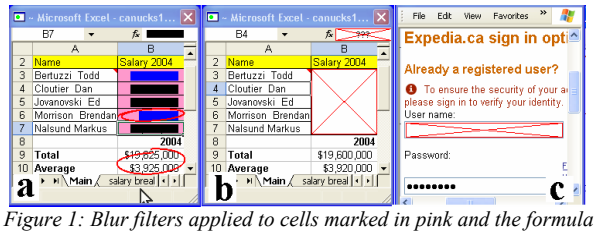

Figure 1: Blur filters applied to cells marked in pink and the formula bar. Auto-circling changes (a,b). Blurring a detected userid field (c).

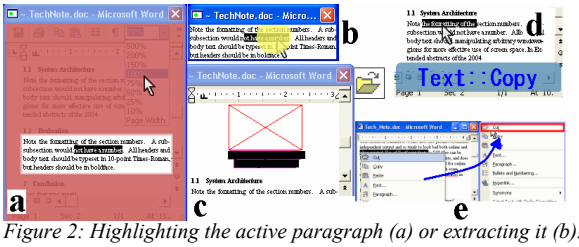

Figure 2: Highlighting the active paragraph (a) or extracting it (b). File open dialog replaced with an icon (c), Replacing a menu with a subtitle(d), Moving an obscuring menu + marking selection (e)