Prefab: What if Every GUI were Open-Source?

Morgan Dixon and James Fogarty
Computer Science & Engineering
DUB Group, University of Washington
{mdixon, jfogarty}@cs.washington.edu

ABSTRACT

Current methods for implementing graphical user interfaces create fundamental challenges for HCI research and practice. Researchers are often unable to demonstrate or evaluate new techniques beyond small toy applications, and practitioners are often unable to adopt methods from the literature in new and existing applications. This position statement examines a vision in which *anybody can modify any GUI of any application*, similar to a scenario where every GUI of every application is open-source. We are currently working to enable this vision through our development of *Prefab*, using pixel-based interpretation of GUIs to enable modification of those GUIs without any cooperation from the underlying application. We see participation in the FLOSS HCI workshop as valuable in at least two regards. First, fully realizing this vision will likely require a community-based approach, so we are interested in Prefab as a platform for collaboration between HCI researchers and the FLOSS community. Second, enabling arbitrary modification of any GUI would seem to blur many current distinctions between open and closed applications, introducing new research questions and further magnifying the importance of the workshop's focus.

Author Keywords

Prefab, pixel-based reverse engineering of GUIs.

ACM Classification Keywords

H5.2. Information interfaces and presentation: User Interfaces.

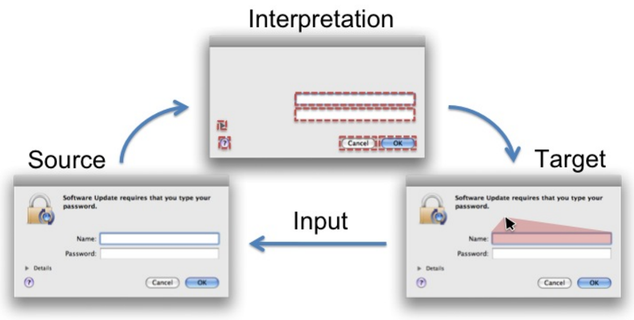
INTRODUCTION

Current methods for implementing graphical user interfaces (GUIs) create fundamental challenges for HCI research and practice. Nearly every GUI is implemented using some form of GUI toolkit (libraries of widgets and associated frameworks to reduce time, effort, and code for implementation), but it is generally difficult or impossible to modify core behavior of these toolkits. A researcher who wants to study a new interaction technique in realistic applications, or a practitioner who wants to adopt a relevant technique from the literature, is therefore generally faced with re-implementing huge portions of an application or toolkit (this is true for not only for closed applications, but also for open applications on closed toolkits). It is both understandable and unfortunate that many researchers instead demonstrate and evaluate new techniques only in small toy applications and that many practitioners limit GUIs to simple combinations of standard widgets [2, 5]. The difficulty of implementing new interaction methods in new and existing GUIs thus limits both the progress of HCI research (i.e., preventing realistic evaluations in complex existing applications) and the broader impact of HCI research (i.e., preventing practitioner adoption of promising state-of-the-art methods in new and existing applications).

This paper discusses a vision in which *anybody can modify any GUI of any application*, similar to the scenario where every GUI of every application is open-source. Realizing such a vision is challenging because (1) we cannot assume application developers will open the source for every GUI, and (2) adding new capabilities to any one application or toolkit has limited impact (people typically use a wide variety of applications built with many different toolkits). We have therefore recently begun to explore an approach based on a commonality of all GUIs: that they ultimately consistent of *pixels* painted to a display. Interpreting GUIs at the pixel level, our Prefab system enables modification of GUIs without requiring cooperation of the underlying application, independent of the tools used to implement those GUIs [1]. We see this as a potential first step in a democratization of the GUI, making it possible for any researcher, practitioner, hobbyist, or end-user to modify and enhance any existing GUI of any application.

PREFAB

Previous approaches to GUI interpretation, such as current accessibility APIs, are based on *how an application is implemented*. Developers expose the functionality of their applications through standard hooks, so enhancements are limited by the (lack of) completeness of these hooks and their implementations. Prefab reverse engineers the structure of GUIs using only their pixels [1]. Prefab is thus based on *what an application looks like*, allowing us to interpret GUIs independent of their underlying application or the toolkit used to implement them. Specifically, Prefab uses a library of *prototypes* to identify the widgets in GUIs. Prototypes can be learned from examples, and can then be shared (i.e., enables a crowdsourcing approach to libraries).



Software Update requires that you type your password.

Figure 1: Prefab interprets GUIs based only on their pixels. Combined with input and output redirection, this can allow anybody to modify any GUI of any application. Here we show an implementation of Grossman and Balakrishnan's Bubble Cursor [3] that operates without

Prefab's pixel-based GUI interpretation can be combined with input and output redirection to allow modification of existing interfaces, as illustrated in Figure 1. In such a process, (1) a bitmap of an existing *source* GUI is captured, (2) the contents of the source are interpreted, (3) a modified GUI is presented in a *target* window (with the source window potentially hidden using virtual desktop methods), (4) input in the target window is mapped back the source, which (5) generates new output that is captured and used to update the target window. The developer of a modification to an existing GUI thus changes the apparent behavior of the source by manipulating the presented target image in combination with the mapping from the target back to the source. Using Prefab and these mechanisms, we have demonstrated the ability to implement several interesting enhancements over a variety of existing applications, including target-aware pointing techniques, Phosphor transitions, and Side View parameter spectrums [1].

PARTICIPATION IN THE FLOSS HCI WORKSHOP

There are at least two aspects of our work on Prefab that we hope will be of interest to the FLOSS HCI workshop.

The first is Prefab as a potential collaboration between HCI researchers and members of the FLOSS community. Prefab is based on libraries of prototypes that describe the appearance of widgets, and long-tail effects (that there are many widgets which appear only in a very small number of applications) suggest a community-driven approach might be effective for building, maintaining, and distributing such libraries. Previous projects have demonstrated potentially relevant approaches (e.g., d.Mix's use of wiki functionality to support inspection and direction modification of scripts web mashup parsing scripts [4]), but significant research questions remain regarding how to effectively design an open community of research and practice around an enabling technology like Prefab. We see participation in the workshop as an opportunity to learn more about effective approaches to interactions between HCI researchers and the FLOSS community, as well as an opportunity to explore the potential for partnerships in developing and studying new approaches to facilitating an open Prefab community.

The second is that Prefab, by allowing anybody to modify GUIs independent of their underlying applications, seems to blur many of the current distinctions between open and closed interfaces. For example, a researcher who wants to evaluate a novel interaction technique in a set of existing applications might create a Prefab-based enhancement and a prototype library corresponding to widgets used in those applications. A practitioner might later polish the implementation of the enhancement for wider distribution, and end-users or hobbyists might in turn extend the prototype library to a much larger number of closed-source applications (perhaps combining the original prototype library with other existing libraries). Different versions of the technique might then emerge as people explore potential modifications of the researcher's original technique. The community has thus effectively added new functionality to a variety of existing closed-source applications. Many of the research questions surrounding open-source interface development seem to be equally important in such a community-driven hybrid of open and closed technologies, and we hope to facilitate discussion of the implications of this approach and additional research questions it suggests.

CONCLUSION

We have briefly introduced our vision of enabling anybody to modify any GUI without requiring cooperation of the underlying application through pixel-based interpretation of existing GUIs together with input and output redirection. We believe the FLOSS HCI workshop would be valuable forum for examining how the HCI research community can collaborate with the FLOSS community around such enabling technology, and we also believe new research questions are suggested by our blurring of the distinction between open and closed interfaces.

ACKNOWLEDGEMENTS

This work was supported in part by Intel and a fellowship from the Seattle Chapter of the ARCS Foundation.

REFERENCES

1. Dixon, M. and Fogarty, J. (2010). Prefab: Implementing Advanced Behaviors Using Pixel-Based Reverse Engineering of Interface Structure. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2010), To Appear.
2. Greenberg, S. and Buxton, B. (2008). Usability Evaluation Consider Harmful (Some of the Time). *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2008), 111-120.
3. Grossman, T. and Balakrishnan, R. (2005). The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2005), 281-290.
4. Hartmann, B., Wu, L., Collins, K. and Klemmer, S.R. (2007). Programming by a Sample: Rapidly Creating Web Applications with d.Mix. *Proceedings of the ACM Symposium on User Interface Software and Technology* (UIST 2007), 241-250.
5. Olsen, D.R. (2007). Evaluating User Interface Systems Research. *Proceedings of the ACM Symposium on User Interface Software and Technology* (UIST 2007), 251-258.