AdaptableGIMP: Designing a Socially-Adaptable Interface

Ben Lafreniere, Filip Krynicki, Michael Terry
HCI Lab, Cheriton School of Computer Science
University of Waterloo, Ontario, Canada
{bjlafren,frkrynic,mterry}@cs.uwaterloo.ca

Andrea Bunt, Matthew Lount
HCI Lab, Department of Computer Science
University of Manitoba, Manitoba, Canada
{bunt,mlount}@cs.umanitoba.ca

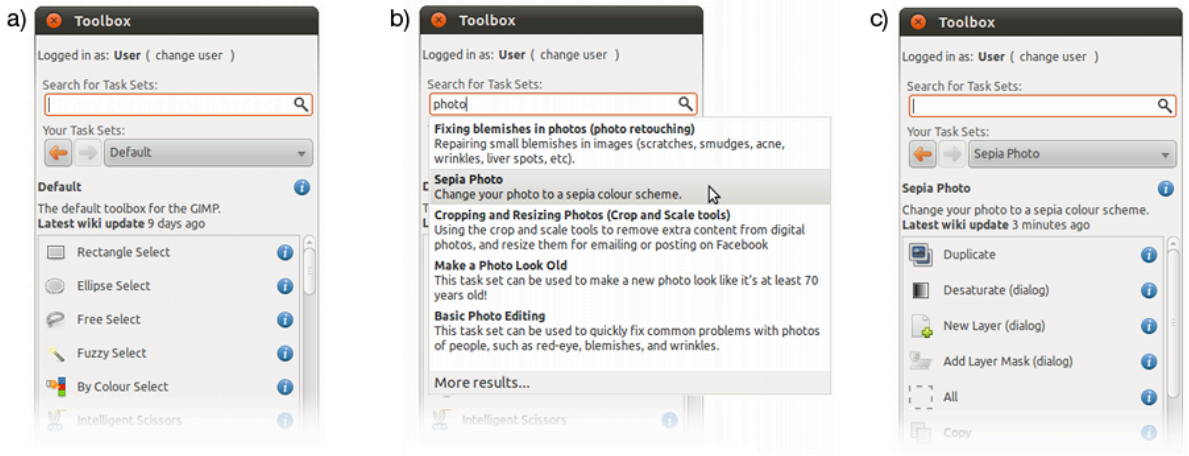

Figure 1. Searching for and installing a task-specific interface customization (a *task set*) in AdaptableGIMP

ABSTRACT

We introduce the concept of a *socially-adaptable interface*, an interface that provides instant access to task-specific interface customizations created, edited, and documented by the application's user community. We demonstrate this concept in AdaptableGIMP, a modified version of the GIMP image editor that we have developed.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors, Documentation

Keywords: Adaptable interfaces, crowdsourcing, wikis, search-based interfaces

INTRODUCTION

Modern desktop applications offer feature sets that far exceed the needs of any one user or task [1]. While a rich feature set makes software applicable to a wide range of problems, it can also lead to decreased efficiency [6], increased frustration when navigating the interface [3], and challenges when learning how to use the software [2].

Adaptable interfaces have been shown to be an effective approach for managing excess interface complexity when well designed [4]. However, new or occasional users of the software may not be familiar enough with the application's functionality to effectively customize the interface. Furthermore, the time spent customizing the interface may yield few benefits if the application is infrequently used.

UIST'11, October 16–19, 2011, Santa Barbara, CA, USA.

We propose a *socially-adaptable interface* approach to help users perform tasks in feature-rich applications. In this approach, the interface provides search-based access to task-specific interface customizations that are created, documented, and shared by the application's user community.

To explore these ideas, we have developed and publicly deployed AdaptableGIMP, a modified version of the GNU Image Manipulation Program. AdaptableGIMP adds the following features and capabilities to GIMP:

- *Task sets*, or interface customizations composed of *ordered sets* of commands and tools (Figure 1c), with accompanying *dedicated wiki pages* for documenting their use and contents (Figure 2)
- A *search-driven interface* for finding and installing task sets from within the application (Figure 1)
- *A centralized task set wiki* for storing and coordinating editing of task sets by the community (Figure 2)
- *Enhanced wiki authoring and change-tracking tools* to accommodate and support interface customization information in each task set's wiki page
- *Dynamically-generated clusters of task sets* on the wiki to support browsing based on community usage (e.g. most installed task sets, most frequently used task sets)

In the most typical use scenario, a user types a few keywords describing what they want to accomplish with the application (Figure 1a), chooses the task set closest to their intended goal (Figure 1b), and performs the task with the resultant streamlined interface (Figure 1c). Documentation

describing how to perform the task can be referenced as necessary (Figure 2).

Authoring a task set involves choosing the commands and tools necessary for the task, naming it, and saving it. When the task set is saved, it is stored as a page on the task set wiki at www.adaptablegimp.org. Once on the wiki, any user can modify the task set's customization details and documentation.

The remainder of this abstract discusses two threads of research that we are exploring with this system: search-based approaches to interface customization, and supporting the crowdsourcing of interface customizations.

A SEARCH-BASED APPROACH TO CUSTOMIZATION

Users often turn to Internet search engines to help them cross the *gulf of execution*—they have a goal that they can succinctly express, but are unsure of how to accomplish that goal in the application [5].

Our search-based approach to customization brings this practice into the application itself. Users' keyword searches return task sets created by the user community, using the rich metadata built around those customizations through community-provided documentation and community usage data.

CROWDSOURCING CUSTOMIZATION

An important component in the design of a socially-adaptable interface is how to support (and encourage) the creation and refinement of customizations by the community. To this end, such a system should:

- Motivate and encourage authoring and improvement of task sets and their documentation
- Provide facilities that enable a loosely coordinated group of users to collectively work on customizations
- Enable refinement of customizations without disrupting current users of these customizations
- Provide guidance regarding what customizations are desired or needed by the community
- Take advantage of implicit community feedback (e.g. logged application usage data, search query logs)

To support community refinement of task sets over time, AdaptableGIMP builds on the rich set of collaborative features provided by MediaWiki (the software on which our wiki is based). Task set customizations are represented as XML text embedded in wiki pages, allowing many of the built-in change-tracking features to seamlessly apply to them. We've further enhanced these features to distinguish between edits to documentation and edits to the list of commands included in a task set.

To lower the barrier to creating and sharing customizations, AdaptableGIMP allows users to create task sets from within the application itself and *automatically* publishes them to the wiki. In a preliminary evaluation, most participants responded positively to this policy, though some were uncomfortable with the idea of other users modifying task sets that they had created.

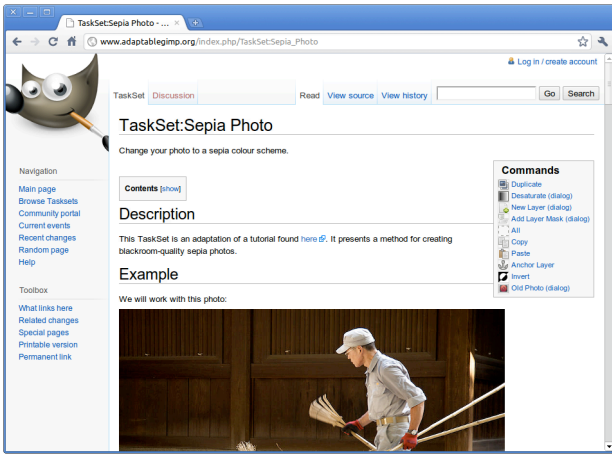

Figure 2. Task set documentation on the AdaptableGIMP wiki

A key design challenge we faced was how to disseminate community refinements of a task set while preserving a stable, predictable user interface for its current users. After exploring a range of alternatives we adopted a model where installing a task set saves a snapshot of its current state at the time of installation, but users can choose to update to the latest community version at any time. To encourage updating, a passive status indicates how recently changes were made to the community version of the currently selected task set.

Finally, inspired by MediaWiki's dynamic "Recent Changes" page, we have created a "Browse Task Sets" page on the wiki that clusters task sets based on a range of real-time community usage data. For example, this page includes a list of task sets ordered by the amount of time the community has spent using them in AdaptableGIMP.

REFERENCES

1. Lafreniere, B., Bunt, A., Whissell, J., Clarke, C.L.A., and Terry, M. Characterizing large-scale use of a direct manipulation application in the wild. *Proc. of GI 2010*, 11-18.
2. Matejka, J., Li, W., Grossman, T., and Fitzmaurice, G. CommunityCommands: Command recommendations for software applications. *Proc. of UIST 2009*, 193–202.
3. McGrenere, J. and Moore, G. Are we all in the same "bloat"? *Proc. of GI 2000*, 187-196.
4. McGrenere, J., Baecker, R.M., and Booth, K.S. A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Transactions on Computer-Human Interaction 14*, 1 (2007), 3.
5. Norman, D.A. and Draper, S.W. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., 1986.
6. Norman, K.L. *The psychology of menu selection: Designing cognitive control at the human/computer interface*. Greenwood Publishing Group Inc., Westport, CT, USA, 1991.