Interactive Systems Architecture Workshop Proposal:

Tools for Generating Procedural Documentation and Content-Adaptive Macros

BJÖRN HARTMANN & MANEESH AGRAWALA
UNIVERSITY OF CALIFORNIA, BERKELEY
EECS DEPARTMENT, COMPUTER SCIENCE DIVISION
{BJOERN,MANEESH}@EECS.BERKELEY.EDU
+1 (415) 868-5720

Research background: Björn Hartmann is an Assistant Professor in Electrical Engineering and Computer Science at the University of California, Berkeley. His research in human-computer interaction focuses on authoring tools for professional designers and end-user programmers. Maneesh Agrawala is an Associate Professor in the same department. He works on visualization, computer graphics and human-computer interaction. His focus is on investigating how cognitive design principles can be used to improve the effectiveness of visual displays.

Travel needs: Support would be appreciated but is not essential. Roundtrip tickets from the San Francisco Bay Area to Jackson, WY cost approximately $500 for the workshop dates.

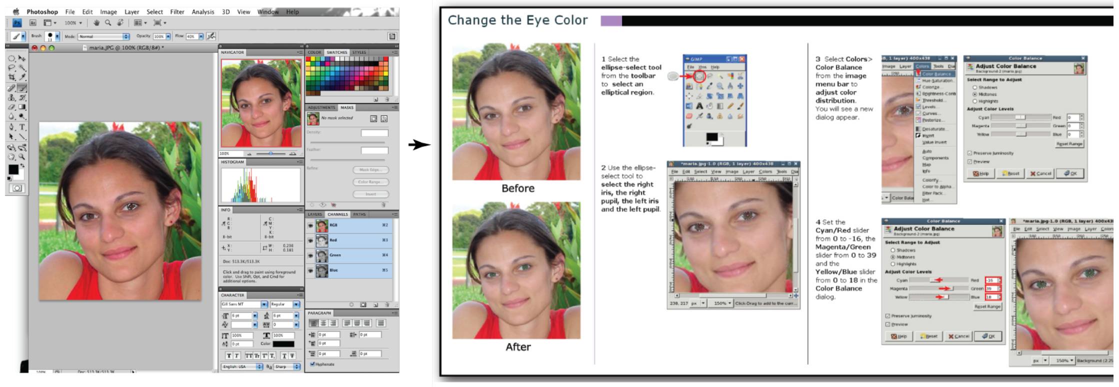
Key research issues: Over the last two decades, we have witnessed a proliferation of software for creating and editing images, audio, video, animation, diagrams, illustrations, 3D models, and user interfaces. Yet the interfaces of such applications can be dauntingly difficult to learn and use effectively. As a result, many users rely on *documentation* to learn how to execute procedural tasks and *macros* to automate repetitive sequences of operations. Supporting application learnability has recently seen a resurgence in research interest (*e.g.*, [3]). However, creating high-quality documentation remains tedious and only a small number of authors have the expertise and patience to create such materials. While many tools allow users to record macros, such macros today are extremely brittle and cannot adapt to new content.

Our goal is to facilitate end-user creation and dissemination of high-quality procedural documentation and macros that robustly adapt to new content. This goal requires both architectural support for capturing work processes and new user interfaces to aid authoring and sharing.

We propose to build a set of tools and techniques that enable users to create and disseminate procedural documentation. Our tools will ***(1) capture*** the interactions and operations users perform as they demonstrate a task and ***(2) generalize*** the specific low-level operations into higher-level actions. From the generalized actions we will develop tools that ***(3) generate*** high-quality tutorials and we will introduce techniques that ***(4) adapt*** these tutorials to new user content. For dissemination, we will ***(1) publish*** the tutorials via the web and application-specific environments, and provide ***(2) search*** tools for finding tutorials using relevance metrics derived from metadata such as problem similarity and popularity. Finally, we will investigate how to enable users to collaboratively ***(3) edit, annotate and debug*** the resulting tutorials.

In pilot research, we are already developing methods that can automatically create photo manipulation tutorials from demonstrations and show suggestions for problems when programming interactive graphics. We recently built an initial prototype system for generating step-by-step photo editing tutorials [1]. While the first results are promising (Figure 1), the tutorials lack semantic explanations that would aid users in transferring knowledge to new problems.

Many digital media artists also write code as part of their work, *e.g.*, to generate graphics procedurally, to author interactive behaviors on web pages, or to create interactive visualizations. We are also developing a tutorial system that aids novices with the debugging of


Change the Eye Color

Before

After

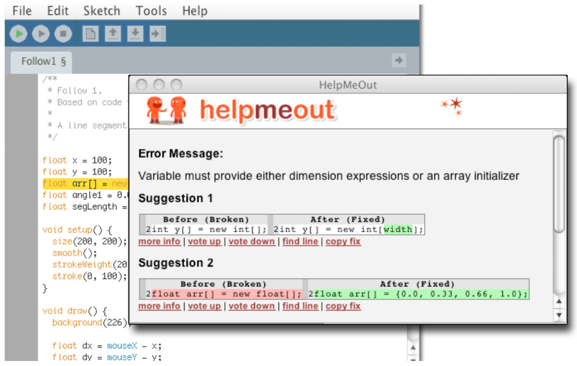
1 Select the **ellipse-select tool** from the **toolbar** to **select an elliptical region.**

2 Use the ellipse-select tool to **select the right iris, the right pupil, the left iris and the left pupil.**

3 Select **Colors> Color Balance** from the **image menu bar** to **adjust color distribution.** You will see a new dialog appear.

4 Set the **Cyan/Red** slider from **0** to **-16**, the **Magenta/Green** slider from **0** to **39** and the **Yellow/Blue** slider from **0** to **18** in the **Color Balance** dialog.

Figure 1: Digital media creation applications like Photoshop (left) are difficult to learn. Our pilot work shows that a demonstration-based approach can be used to generate photo manipulation tutorials (right). We will significantly extend our approach to produce much higher quality tutorials.



Error Message:

Variable must provide either dimension expressions or an array initializer

Suggestion 1

Before (Broken)	After (Fixed)
`2int y[] = new int[];`	`2int y[] = new int[width];`

more info | vote up | vote down | find line | copy fix

Suggestion 2

Before (Broken)	After (Fixed)
`2float arr[] = new float[];`	`2float arr[] = {0.0, 0.33, 0.66, 1.0};`

more info | vote up | vote down | find line | copy fix

Figure 2: Novices who use graphics programming environments like Processing often have significant difficulty interpreting errors messages. Our work shows that suggestions for correcting errors (shown above) can be generated automatically by recording work sessions of other programmers.

compiler error messages and runtime exceptions for such programs by suggesting successful solutions to similar errors that other programmers have encountered (Figure 2, [2]).

This research will increase the quality of procedural documentation and macros available to end-users and thereby facilitate learning and use of complex design software, especially for novices and amateurs. We believe that web-based access to such materials will further democratize the production and sharing of design knowledge. At a deeper level, our research produces three types of results: 1) knowledge about how people learn to use complex software, 2) algorithmic techniques that leverage that knowledge to produce high-quality documentation from demonstrations, and 3) novel approaches for automatically adapting macros to work with new content.

References:

1. Grabler, F., Agrawala, M., Li, W., Dontcheva, M., and Igarashi, T. 2009. Generating photo manipulation tutorials by demonstration. In *ACM SIGGRAPH 2009 Papers* (New Orleans, Louisiana, August 03 - 07, 2009). H. Hoppe, Ed. SIGGRAPH '09. ACM, New York, NY, 1-9.

2. Hartmann, B., MacDougall, D., Brandt, J., and Klemmer, S. R. 2010. What would other programmers do: suggesting solutions to error messages. In *Proceedings of the 28th international Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA, April 10 - 15, 2010). CHI '10. ACM, New York, NY, 1019-1028.

3. Matejka, J., Li, W., Grossman, T., and Fitzmaurice, G. 2009. CommunityCommands: command recommendations for software applications. In *Proceedings of the 22nd Annual ACM Symposium on User interface Software and Technology* (Victoria, BC, Canada, October 04 - 07, 2009). UIST '09. ACM, New York, NY, 193-202.