GRAPHICAL EDITING BY EXAMPLE

David Kurlander

Department of Computer Science
Columbia University
New York, NY 10027

E-Mail: djk@microsoft.com

INTRODUCTION

Graphical editing, like many applications facilitated by computers, often involves repetitive tasks. To reduce repetition, programmers can write procedures to automate these tasks, however most users do not know how to program, and the repetitive tasks that they perform are frequently too specialized for the application programmer to anticipate. End users would benefit from the ability to customize and extend their applications for the tasks they usually perform.

Programming by example systems and demonstrational interfaces aim to give end users this capability. Such systems are programmed simply by using the applications, rather than through an ancillary extension language. Innovative systems such as Pygmalion, Tinker, SmallStar, Peridot, Metamouse, and Eager have all explored ways of bringing more power to the non-programming end user [1]. The accompanying videotape demonstrates Chimera, a system built to explore new demonstrational techniques in the domains of graphical editing and interface building.

DESCRIPTION

Chimera employs five novel demonstrational techniques to make such editing easier. While most of these techniques are independently useful, they also work together to make a more powerful system. The five techniques demonstrated in the videotape are summarized below:

Graphical search and replace is the graphical analogue to textual search and replace in text editors, and is useful for making repetitive changes in graphical documents [2]. Search patterns can include graphical attributes such as fill color and line style, and searches can be performed on shape as well, optionally matching objects at arbitrary rotations and scales. Graphical replacements can alter the same or different set of graphical attributes and shape.

Constraint-based search and replace extends graphical search and replace to match on geometric relationships, not just complete shape [5]. Constraints in the search pattern specify relationships to look for, and constraints in the replacement pattern indicate which relationships to change, as well as which to hold constant.

Constraints from multiple snapshots allows constraints to be specified by demonstration [4]. The user provides snapshots of multiple valid configurations of a scene, and the system automatically determines which constraints are valid in all of them. The prior technique can be used to infer constraints from a *static* scene, however this technique can infer constraints from a *dynamic* scene, which contains more information.

Editable graphical histories follow a comic strip metaphor to display graphical user interface commands in a series of panels [3]. Related operations are coalesced in the same panel, and each panel shows only those scene objects participating in the operation, plus a little scene context. These panels are useful for reviewing the history, undo and redo.

Graphical macros by example allow the user to, at any time, scroll through an editable graphical history, and encapsulate useful sequences of commands into macros [6]. These macros can be parameterized and generalized to work in other contexts. They can use graphical and constraint-based search as flow of control mechanisms.

ACKNOWLEDGMENTS

My advisor, Steven Feiner, helped direct this research. Eric Bier suggested the problem of graphical search and replace, and worked with me on that component.

REFERENCES

1. Cypher, A. *Watch What I Do: Programming by Demonstration.* To appear 1993.
2. Kurlander, D. and Bier, E. A. Graphical Search and Replace. Proceedings of SIGGRAPH ‘88. In *Computer Graphics 22*, 4 (August 1988). 113-120.
3. Kurlander, D. and Feiner, S. A Visual Language for Browsing, Undoing, and Redoing Graphical Interface Commands. In *Visual Languages and Visual Programming*, S. K. Chang, ed. Plenum Press, NY 1990. 257-275.
4. Kurlander, D. and Feiner, S. Inferring Constraints from Multiple Snapshots. Columbia University Tech. Report CUCS-008-91. May 1991.
5. Kurlander, D. and Feiner, S. Interactive Constraint-Based Search and Replace. In *CHI ‘92 Proceedings.* May 1992. 609-618.
6. Kurlander D. and Feiner, S. History-Based Macros by Example. To appear in *UIST ‘92 Proceedings.* November 1992.

---

*Author’s current address: Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399.