

Software for Systematic and Imaginative Exploration

Daryl H. Hepting

Department of Computer Science

University of Regina

Regina, S4S 0A2, Canada

dhh@cs.uregina.ca

ABSTRACT

This paper describes a software system that has been designed to enhance individual creativity. It was once thought that the era of information visualization would empower people and enable a democratization of visual thinking. Today, many barriers to this democratization still exist. The software, called *cogito*, is designed from the perspective that it is possible, with appropriate support, to meaningfully explore a very large set of alternatives without becoming overwhelmed. This paper discusses the design choices made for the software, the architecture of the system, and the experience gained in a variety of applications.

Author Keywords

Interaction, training, creativity, software design

ACM Classification Keywords

H5.2. Information interfaces and presentation (e.g., HCI): Interaction styles (e.g., commands, menus, forms, direct manipulation); Training, help, and documentation

INTRODUCTION

Jacques Bertin[1] wrote that “to construct 100 DIFFERENT FIGURES from the same information requires less imagination than patience.” Selecting amongst these variations requires judgement, which may differ between users. Many variables, including task[3], can affect this judgement. Amongst the selected alternatives, one might find something that is novel and unexpected. Boden[2] distinguished between H-creativity (historical) and P-creativity (personal), depending on whether the item is new to society or only new to the individual. Both, or neither, may apply to any one alternative.

Some alternatives, according to Gero[5], will be exceptional while others will be routine. Shneiderman[10], with his *genex* framework, would direct computational support towards creation of the exceptional, and possibly H-creative, designs. However, P-creativity is important because it means that the individual is inventing. When the 100, or 1,000,000, alternatives are available in a software application, the user

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

C&C'07, June 13–15, 2007, Washington, DC, USA.

Copyright 2007 ACM 978-1-59593-712-4/07/0006...\$5.00.

must become an expert in the software tool in order to access those alternatives. However, it may be difficult to maintain expertise in all the facets of the software's operation required to realize solution alternatives. And, immersion in the command syntax of a tool may alter one's semantic view of the problem at hand. Patience is a virtue that is in decline in modern society. If software could be designed to reduce Bertin's requirement of the user for patience, could it help to democratize creative visual thinking for users by enabling more imagination? Such software could lower the threshold so that more citizens could be more creative, and thus provide an answer to the question of how to cultivate and sustain creativity.

BACKGROUND

Licklider[8] first talked about a human-computer symbiosis in which computers augmented people's own capacities. Kocchar et al.[7] describe three ways in which a user might work with a computer. *Manual* systems put too much responsibility on the user and *automated* systems take too much responsibility away from the user. *Augmented* systems view the computer as a contributing partner[4]. In this mode, the user and the computer share tasks based on the abilities of each.

Sims[11] presented a software system that allowed people to experiment with the creation of computer graphical images using an evolutionary process. This system allowed the user to focus on the end product rather than on the means to generate each particular image. The Design Galleries[9] system addresses many of the same issues outlined here, but it is limited by the need for the user to describe, *a priori*, how the system should differentiate between alternatives. The Side-Views[12] system promotes exploration of command alternatives but it does so from within an application by creating previews attached to command menus. The *flamenco* search interface[13] works with faceted metadata that describes the features of the problem, but the display structure does not allow the user to see exemplars until the last stages of navigation.

DESIGN

The approach taken with *cogito*[6] allows the user to explore the space of alternatives available from an application in ways that are both systematic and imaginative. The user begins by conceptualizing a problem and describing it to the *cogito* system. Figure 1 shows an exploration of simple curves using GNUplot. Imagine that the curves of interest

have 4 control points, the first at (0,0) the last at (1,1) and the middle two points at $(1/3, y_1)$ and $(2/3, y_2)$ respectively. If y_1 and y_2 are each allowed to take on 9 different values in the range (0,1), and the curve can use one of bezier, cspline, or linear interpolation, there are immediately 243 alternatives to consider. Systematic exploration is enabled by the user examining, in sequence, the effect of different values for y_1 on the rendered curve. Imaginative exploration is enabled by allowing the user to select curves of interest and then to perform a genetic cross-over on them to see new alternatives. The bookkeeping of this exploration is handled by *cogito*. An exhaustive examination of these alternatives would require 30 more screens like Figure 1.

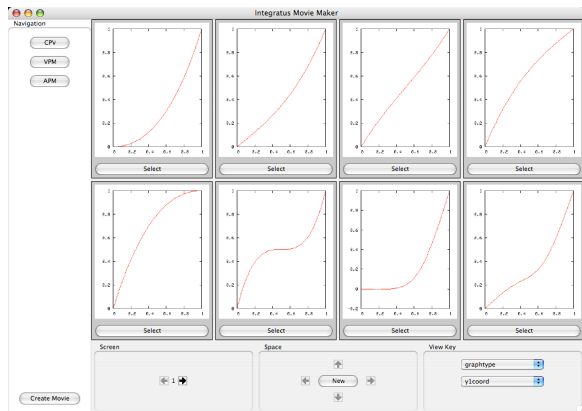


Figure 1: Exploration of simple curves using GNUplot.

The *cogito* architecture is such that it can act as an interface to a variety of applications, so long as they provide some form of batch interaction. One such application is the statistical package called R¹, which is not known for its ease of use. Figure 2 shows *cogito* being used to explore the output from R. Each cell shows a different visual representation, which quickly gives a sense of the breadth of options available within R. This shows promise as a way to train users in a new application: the complete capabilities of an application need not be understood before this interface can be attached and the application can be explored.

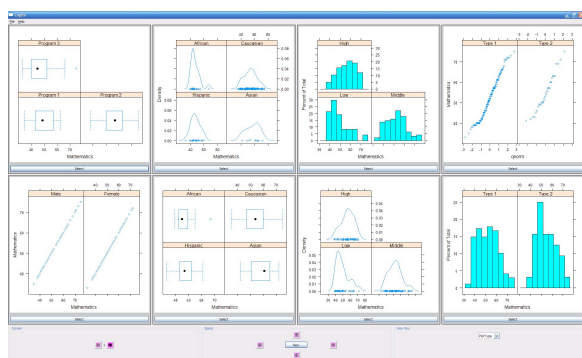


Figure 2: Exploration of visualizations using R.

The interface is inspired by Sims' work. Unlike Design Galleries, nothing must be specified *a priori*. This approach may be helpful in discovering the objective function that could be used in Design Galleries. Unlike Side-Views, the user does not have to deal with the intricacies of any particular application because this interaction is done outside of any particular application. And, unlike flamenco, the user is always able to see exemplars to guide his or her exploration. Development of *cogito* continues, but already its features have encouraged exploration within applications.

ACKNOWLEDGEMENTS

Paul Schmiedge and Matthew McKague have made outstanding contributions to this software. This work was funded by the Natural Sciences and Engineering Research Council of Canada. I thank Jennifer Hepting, Timothy Maciag, and the anonymous reviewers for their comments.

REFERENCES

1. Bertin, J., *Semiology of graphics: diagrams, networks, maps*. Univ. Wisconsin Press, 1983. Trans. W. J. Berg.
2. Boden, M. A., Creativity. In M. A. Boden, ed., *Artificial Intelligence: Handbook of Perception and Cognition*, Chapter 9. Academic Press, 2nd Ed. 1996.
3. Casner, S. M. A task-analytic approach to the automated design of graphic presentations. *ACM TOG*, 10(2), 1991.
4. Engelbart, D. C., A conceptual framework for the augmentation of man's intellect. In Howerton and Weeks, eds., *Vistas in Information Handling*, Volume 1, pages 1–29, Spartan Books, London, 1963.
5. Gero, J. S. Computers and Creative Design, *CAAD Futures*, 1995.
6. Hepting, D. H. and Schmiedge, P., A User-Driven Interface for Exploring Visualizations, *Proceedings of Visualization and Data Analysis (VDA) 2007*, 2007.
7. Kochhar, S., J. Marks, and M. Friedell. Interaction paradigms for human-computer cooperation in graphical-object modelling. In S. MacKay and E. M. Kidd, eds., *Proc. Graphics Interface '91*, 1991.
8. Licklider, J. Man-computer symbiosis. *IRE Trans. Human Fact. Elect.*, 4–11, 1960.
9. Marks, J. et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Computer Graphics: SIGGRAPH Proceedings*, 1997.
10. Shneiderman, B. Creating Creativity: User Interfaces for Supporting Innovation, *ACM TOCHI*, 7(1), 2000.
11. Sims, K.. Artificial evolution in computer graphics. In R. J. Beach, editor, *Computer Graphics: SIGGRAPH '91 Conference Proceedings*, pages 319–328, 1991.
12. Terry, M. and E. D. Mynatt, Supporting experimentation with Side-Views, *CACM* 45(10), 2002.
13. Yee, K.-P., K. Swearingen, K. Li and M. Hearst, Faceted metadata for image search and browsing. *Proceedings ACM SIGCHI*, pages: 401 – 408, 2003.

¹ <http://www.r-project.org/>