

SageTools: An Intelligent Environment for Sketching, Browsing, and Customizing Data-Graphics

Steven F. Roth, John Kolojejchick, Joe Mattis, Mei C. Chuah

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
+1 (412) 268-7690
Steven.Roth@cs.cmu.edu

ABSTRACT

Our approach views data-graphic design as two complementary processes: design as a constructive process of selecting and arranging graphical elements, and design as a process of browsing and customizing previous cases. We present three novel tools for supporting these processes. *SageBrush* assembles data-graphics from primitive elements like bars, lines, and axes. *SageBook* browses previously created data-graphics relevant to current needs. *SAGE* automatically designs data-graphics, interpreting the user's specifications as conveyed with the other tools. The combined environment, SageTools, enhances *user-directed* design by providing automatic presentation capabilities with styles of interaction that support data-graphic design.

KEYWORDS: Graphic Design, Data Visualization, Automatic Presentation Systems, Intelligent Interfaces, Design Environments, Interactive Techniques

MOTIVATION

Data-graphics (e.g., line charts, bar graphs, tables, networks, etc.) provide valuable support for data exploration, analysis, and presentation, but the task of designing effective data-graphics can require considerable expertise. A number of software packages exist for generating data-graphics, but these packages have several significant limitations. First, the interfaces to these packages typically consist of menus that enumerate a bewildering number of alternative styles and options, and can be time-consuming both to learn and to use. Second, these packages often design ineffective data-graphics, because the packages cannot produce *integrative* displays for viewing the relations among several data attributes or data sets, and instead force the user to isolate each data set in a separate data-graphic. Finally, these packages provide little guidance for the majority of users who are not experienced graphic designers.

One approach to solving these problems is to build systems with built-in graphical design knowledge, which enables

the system to select and design effective displays based on descriptions of the data and the goals of the viewer [1,2,3,5]. This research has provided a vocabulary for describing the elements of data-graphics, along with knowledge of the appropriateness of these elements to different data and tasks, and has identified the design operations for combining these elements to form integrative, unified displays.

Armed with this knowledge, automatic design systems should reduce the need for interaction and expertise, while providing great flexibility in customizing displays. However, previous automatic design research has not been concerned with supporting user interactions, and has instead focused only on the issues of identifying and encoding knowledge of data, tasks, and design. No paradigms have been developed for a collaborative process between human and automated designers. This video shows a novel approach to interactive graphic design, which has been implemented as a software suite called SageTools. With SageTools, automatic mechanisms can be used to support users, and not replace them, in the graphic design process.

OVERVIEW

Our approach to supporting design has been to integrate an evolving automatic presentation system called SAGE [6] with two new interactive design tools called SageBrush and SageBook. Both tools enable users to perform natural design operations by manipulating familiar objects, shielding users from the complex internal representations and operations that SAGE uses to create graphics.

SageBrush is representative of design tool interfaces in which users specify graphics by constructing *sketches* from a palette of primitives and partial designs. SageBrush provides a flexible, generative, direct manipulation design interface, in which users assemble a large number of possible combinations of graphical elements (e.g. axes, lines, text, etc.) by performing simple drag-and-drop operations. Users can customize the spatial and structural relationships among the different graphical elements, and map these elements to the data that they wish to visualize. Each sketch serves as a schematic view of a graphical design that is translated into a set of *design directives*, which are specifications expressed in SAGE's graphic representation language. Design directives from

To appear in
Proceedings CHI'95
Human Factors in
Computing Systems,
ACM, May 1995.

SageBrush guide SAGE's automatic processes, and provide criteria for SageBook to use in searching its library of previously designed pictures.

SageBook is an interface for browsing and retrieving previously created data-graphics and utilizing them to visualize new data. SageBook supports an approach to design in which people remember or examine previous successful visualizations and use them as a starting point for designing displays of new data, extending and customizing them as needed. Our experiences in graphic design suggest that search and reuse of prior cases with customization is a common process. Therefore, our goal is to provide methods for searching through previously created pictures based on their graphical properties and/or the properties of the data they express. Once a suitable design is found, the user can optionally modify it using SageBrush, and then send the design to SAGE, which will use it to create a visualization of the new data.

Users direct SageBook in its searches by performing operations in SageBrush. By sketching graphical primitives, the user directs SageBook to search for data-graphics that resemble the sketch. By selecting data sets, the user directs SageBook to search for data-graphics expressing data that is similar to the selected data. Additionally, users can specify different degrees of relaxation on the search criteria, ranging from highly constrained exact matching to more relaxed partial matching. Exact matching can serve users who have a good idea of what data-graphics they need (e.g., generating a monthly report in the same format as before). Partial matching can serve users who need guidance in creating a new, customized design (e.g., locating a good starting point when generating a report for the first time).

SAGE is an automatic presentation system containing many features of related systems like APT, BOZ, and ANDD [1,3,4]. SAGE takes as input a characterization of the data to be visualized, along with the user's data viewing goals. Design operations include selecting graphical techniques based on expressiveness and effectiveness criteria, and composing and laying out graphics appropriate to data and goals. SAGE can create data-graphics when users completely specify a design in SageBrush, as well as when users provide no specifications at all. Most importantly, SAGE can accept *partial* specifications at any level of completeness between these two extremes and finish the design reasonably. User specifications serve as *design directives*, which constrain the path of a search algorithm that selects and composes graphics to create a design.

The ability to accept partial specifications from SageBrush is enabled by a rich *object representation* of the components of graphic displays, including their syntax (i.e., their spatial and structural relationships) and semantics (i.e., how they indicate the correspondence

between data and graphics). The representation allows SAGE to produce combinations of a wide variety of 2D graphics (e.g. charts, tables, map-like coordinate systems, text-outlines, networks). The representation also enables SAGE to support SageBook in its searches for previous data-graphics that have particular graphical elements specified by users.

CURRENT WORK

We are currently extending SageTools so that every data-graphic becomes an interface for manipulating data, allowing the user to produce effective visualizations of large data sets. Utilizing dynamic query techniques, SageTools can provide sliders and selectors that enable users to control the visibility of data-elements. Painting and brushing techniques can allow users of SageTools to coordinate data-elements across multiple regions of a picture, thus enhancing SageTools' integrative capabilities. Aggregation and decomposition techniques can allow users to control the level of detail at which data is presented. Finally, new visualization techniques, such as animation and three-dimensional representation and navigation, can enhance the power of SageTools' graphical repertoire.

By combining the use of a knowledge-based automatic presentation system, a direct manipulation constructive interface, and a content-based browser, SageTools provides a paradigm for collaboration between human and automated designers. This paradigm supports the complementary processes of constructing data-graphics by selecting and arranging graphical elements, and of constructing data-graphics by browsing and customizing previous cases, illustrating the power of using such an interdisciplinary approach.

REFERENCES

1. Casner, S. M. A Task-Analytic Approach to the Automated Design of Information Graphic Presentations. *ACM Transactions on Graphics*, 10, 2 (Apr. 1991), pp. 111-151.
2. Mackinlay, J. D. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*., 5, 2 (Apr. 1986), pp. 110-141.
3. Marks, J. W. Automating the Design of Network Diagrams. Ph.D. thesis, Harvard University, 1991.
4. Roth, S.F., Kolojechick, J., Mattis, J., and Goldstein, J. Interactive Graphic Design Using Automatic Presentation Knowledge. *Proceedings SIGCHI'94 Human Factors in Computing Systems*, Boston, MA, ACM, April, 1994, pp. 112-117.
5. Roth, S. F. and Mattis J. Data Characterization for Intelligent Graphics Presentation. *Proceedings SIGCHI'90 Human Factors in Computing Systems*, Seattle, WA, ACM, April, 1990, pp. 193-200.
6. Roth, S. F. and Mattis, J. Automating the Presentation of Information. *Proceedings IEEE Conference on AI Applications*, Miami Beach, FL, Feb. 1991, pp. 90-97.