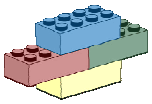

A web based architecture for visually coordinating and publishing multiple-view visualizations

Nathan Conklin
nconklin@cs.vt.edu
http://infovis.cs.vt.edu/snap/
Center for Human-Computer Interaction & Department of Computer Science
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061

ABSTRACT

User coordinated visualizations are a powerful mechanism for building a visual interface. They allow a user to tailor an interface to their needs by choosing components, allowing visual representations customized to a task or user preferences. Snap-Together Visualization (Snap) is a web based system that allows for the creation and sharing of user customized visual interfaces. Users can bring their data to the system, compose a multiple-view interface, and visualize the data. They are then able to share the data and visual interface with other web users by providing a single URL. Snap provides a flexible architecture for coordinating independent interface components and sharing the resulting visualization.

Keywords

information visualization, multiple views, software architecture, component coordination, web based sharing

INTRODUCTION

Multiple-view visualizations are an effective medium for users to explore complex information spaces. These visualizations provide users with different representations of a single conceptual entity, allowing improved performance and the discovery of unforeseen relationships [1, 2, 3, 4]. However, these visualizations are typically composed of specific components with fixed methods of interaction. Users do not have the ability to choose how they want to visualize a dataset or how one view is coordinated with another.

Users need the flexibility to coordinate visualizations in ways unforeseen by developers. Developers should have the freedom of not worrying about the required coordination management and event propagation for such a system. Previous work introduced a Windows based prototype that allowed users to build their own visualization by integrating preexisting visualization components [2]. It required a significant installation base and required rebuilding to add new visualization components. The current research extends Snap-Together Visualization (Snap). It introduces a new architecture and user interface. The architecture allows for run-time visualization additions and requires no footprint. It also provides users with the capability to publish the visualizations that they have constructed. The new user interface allows for visually coordinating the visualization components.

SNAP-TOGETHER VISUALIZATION

Snap is a continuing project that supports user construction and coordination of multiple-view visualizations. It is a web based system that allows users to construct visualizations by combining shared components that have been submitted by developers. It builds on the existing model for coordinating visualizations based on relational schemata [4].

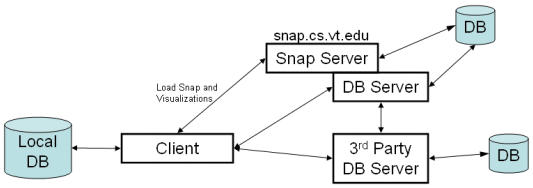

Figure 1: Snap system architecture

As shown in Figure 1, users can connect to either local or remote databases and interact with the data using visualizations that are registered with the Snap server. The Snap system architecture supports visualization component submission, user-based coordination, and sharing of custom-built visualizations.

Software Architecture

The software model is an event based, implicit invocation architecture. This allows component developers to be concerned only with firing and receiving events and not with explicitly coordinating with other components [5]. Snap builds on the existing coordination architecture [2] by allowing the association of actions with multiple objects.

The coordination manager utilizes the relational schemata to convert related keys when firing events. The architecture also allows for a middle layer to communicate with visualizations that utilize differing technologies. Figure 2 presents a diagram of the software components that make up the Snap architecture.

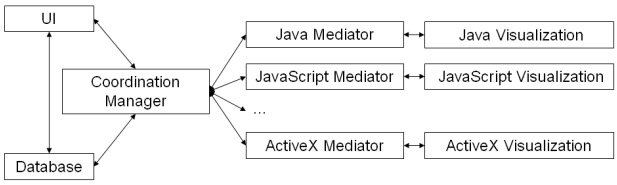

Figure 2: Snap software component architecture

Coordination Interface

The Snap project is investigating ways to visually coordinate visualization components. Two approaches that are being considered include a diagrammatic visualization and a datacompass [2]. The diagrammatic visualization allows visualizing the component coordinations as a network graph diagram. The datacompass allows for visually navigating the relational schema and assigning visualizations within the navigation.

Careful consideration is given to how to use multiple views [1]. Layout design uses a space filling approach for presenting the visualization components. Visual coordination helps make component relations apparent. Bi-directional event firing maintains consistency in view state [3]. Snap is also investigating techniques to focus user attention on the relevant view when appropriate.

Visualization Publishing

The Snap system allows a user to save the layout that they have constructed. This saved layout includes the database connection details, the visualization choices, and the user defined coordination. This layout can be reopened using a single URL. When a user opens this URL, the Snap system connects to the database, loads the appropriate visualizations, and reestablishes the saved coordinations. This allows a user to publish their customized multiple-view visualization by publishing the URL.

Saving a layout also saves each visualization's state. Using event playback, Snap can reestablish each visualization to its previously saved state. This technique provides for asynchronous collaboration, allowing users to share the insight that they gained while interacting with the visualization.

FUTURE WORK

Development work is currently being pursued for the architecture, coordination interface, and publishing capabilities. Future directions include evaluating the different coordination interfaces, tight coupling of the coordination manager with the component frames, and enabling a single location for the management of different visual encodings for the primary and secondary views.

ACKNOWLEDGMENTS

Thanks to Chris North for his vision and support in this research.

REFERENCES

1. Baldonado, M., Woodruff, A., Kuchinsky, A. "Guidelines for Using Multiple Views in Information Visualization", *Proc. ACM Advanced Visual Interfaces 2000*, pp. 110-119, 2000.
2. North, C. "A User Interface for Coordinating Visualizations Based on Relational Schemata: Snap-Together Visualization", *University of Maryland, Computer Science Dept.*, Doctoral Dissertation, 2000.
3. North, C. "Multiple Views and Tight Coupling Visualization: A Language, Taxonomy, and System", *Proc. CSREA CISST 2001 Workshop of Fundamental Issues in Visualization*, pp. 626-632, 2001.
4. North, C., Shneiderman, B. "Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata", *Proc. ACM Advanced Visual Interfaces 2000*, pp. 128-135, 2000.
5. Shaw, M., Garlan, D. *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.

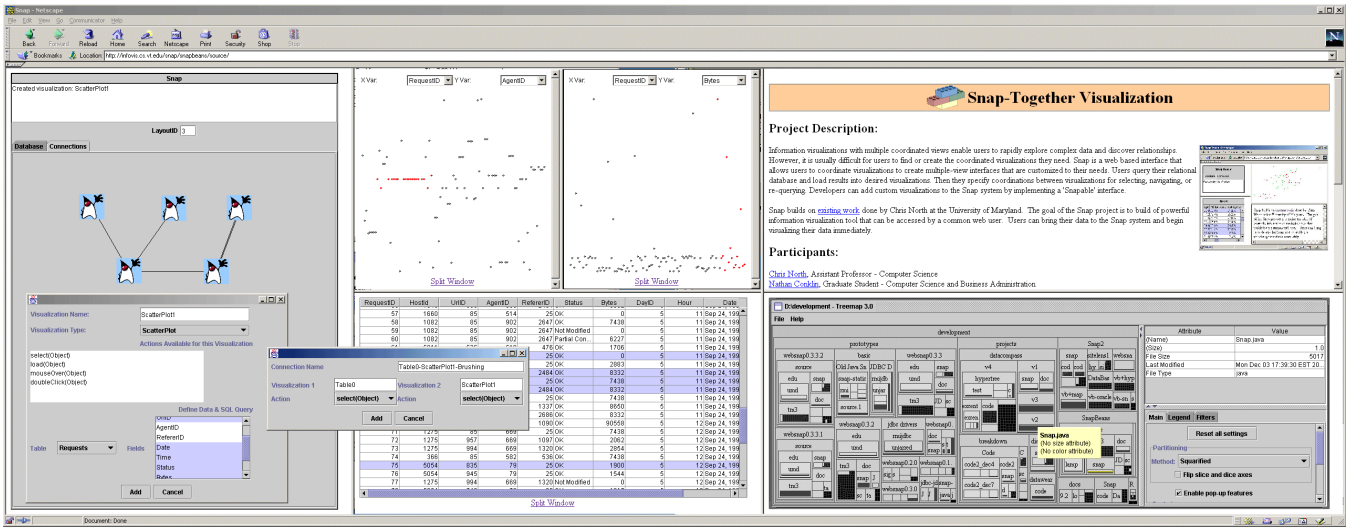

Figure 3: Snap is a web based system that enables users to publish their customized visualizations using a single URL.