VisBubbles: A Workflow-Driven Framework for Scientific Data Analysis of Time-Varying Biological Datasets (SAP-0281)

Guangxia Li* Andrew C. Bragdon† Zhigeng Pan‡ Mingmin Zhang§
Zhejiang University Microsoft Research Zhejiang University
Sharon M. Swartz¶ David H. Laidlaw‖ Chaoyang Zhang** Hanyu Liu†† Jian Chen‡‡
Brown University University of Southern Mississippi

Abstract

We present VisBubbles, a unified environment supporting programming, visualization, and interaction concurrently for data analysis workflow. A key aspect of VisBubbles is its extension of static multiple views to a metaphorical interface of bubbles that becomes a flexible layout to support analysis, motivated by the recent success of integrated development environment in Code Bubbles [Bragdon et al. 2010]. Another key aspect is the pipeline approach that combines programming and interaction in such a way that newly authorized data can be visualized without leaving the data analysis environment. The framework is developed in a participatory design process and discussion with biologists continues to provide new insights into general-purpose support for data-analysis workflow.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques;

Keywords: Workflow, scientific visualization, motion analysis

Links: DL PDF

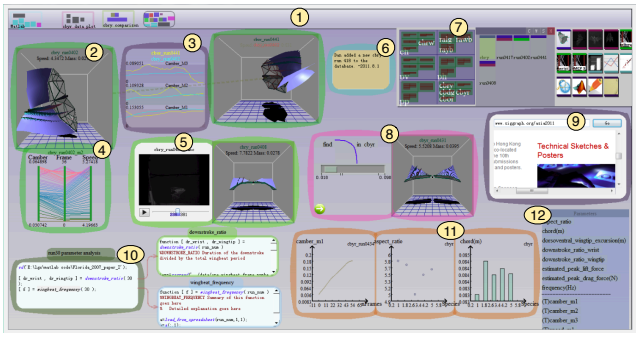

Figure 1: *VisBubbles interface. An example visual interface that a biologist can build using VisBubbles, which contains (1) a virtual work space and the following bubbles: (2) geometry, (3) comparison plot, (4) parallel coordinates, (5) video, (6) note-taking, (7) data source manager, (8) sketch bubbles for shape fitting, (9) web bubbles, (10) Matlab programming bubbles, (11) 2D plots, and (12) authorized data.*

1 Introduction

Scientists spend a significant amount of time analyzing their experimentally captured data; this data analysis workflow process, starts when the data are being collected and ends when scientific insights are achieved. This analytical process entails programming for feature extraction and visualization and interaction to confirm analytical results. Our long-term collaborations with bat flight biologists have revealed that several workflow issues continue to be persistent barriers in scientists' data analysis tasks, because the process involves frequent feature extractions, visualizations, and comparative studies of heterogenous datasets. Conventional problem-solving environments have advanced workflow design by focusing on provenance-aware systems and automating repetitive tasks to ensure worthwhile results. However, only limited attention has been given to user interfaces that bridge programming and visualization in environments involving multiple information resources, feature extraction by programming, and analytic study. We advocate a novel user interface design for editing and visualizing multiple artifacts in the analysis process to support multiple tasks without forcing the scientists to exit from the problem solving environments.

*e-mail: liguangxia@zjucadcg.cn
†e-mail:acb@cs.brown.edu
‡zgpan@cad.zju.edu.cn
§zmm@cad.zju.edu.cn
¶e-mail:sharon_swartz@brown.edu
‖e-mail:dhl@cs.brown.edu
**e-mail:chaoyang.zhang@usm.edu
††e-mail:hanyu.liu@eagles.usm.edu
‡‡e-mail:jian.chen@usm.edu

2 Related Work

Workflow environments seek to streamline analytical processes in scientific visualizations so as to investigate scientific problems, help propose new hypotheses, and compare results [Callahan et al. 2006]. VisTrails contributes to the infrastructure built on VTK and turn the traditional *ad-hoc* data visualization process into a pipelined approach to automate repetitive visualization tasks by providing detailed history information [Callahan et al. 2006]. The Advanced Visual System (AVS) [Upson et al. 1989], SCIRun [Parker and Johnson 1995], and ParaView (http://www.paraview.org) execute variations of a given pipeline with different parameter inputs. The users can modify the parameters manually via the interface and see the results. These workflow environments addresses difficult problems within the visualization process alone but omit the programming part, which is crucial in processing large datasets for analysis even before visualization can be used.

3 Design Analysis

Our long-term collaboration with bat biologists has revealed three workflow design issues. The first is that **visual interfaces have not provided enough support in the dynamic data analysis process**. The biologists' analytic process is dynamic, not static. That the path of their exploration is often unpredictable imposes several design requirements. They need easy access to a tremendous amount of data to reason through their analysis, and they need an interface that supports their multiple diverse and simultaneous tasks. Thus, an interface is needed that **adapts to different analysis stages for rapid refinement of analytical tasks and hypotheses**.

The second issue is that **program implementation and data analysis are conducted in multiple separate working environments**. Evolutionary biologists use Matlab to conduct analysis and then program the results into visualizations for confirmation or generation of new hypotheses. If more analysis is needed, they switch back to Matlab to make changes. Switching back and forth between Matlab and the visualization process introduces significant context switching consts. The visual interface must **scale to the application environment** so that data analysis can be conducted in a single pipeline and the user is freed from bridging the gaps.

The third issue is that the **current implementation of visualizations forces the users to choose the encoding approach**. Selecting the right mappings from the data to the visual components is difficult, especially when the data are heterogeneous. Our collaborators are reluctant to handle this step, greatly preferring to focus on their analytical tasks alone. Some abnormalities in their data sets that would have been caught in visualization were instead detected in a much later analysis stage, with significant loss in productivity. One solution is that the visual interface design should **provide some good defaults suitable to the data at hand**.

4 Our Methods and Preliminary Results

We addressed these issues in the design of VisBubbles (Fig. 1), which was inspired by two characteristics of human discovery as a decision making process. The first was inspired by James's two-stage decision making theory [James et al. 1975]; this describes a temporal sequence of undetermined alternative possibilities followed by adequately determined choices and thus precisely addresses the dynamic workflow issue laid out above.

In a nutshell, the opportunistic stage reflects the hypothesis generation stage of looking at all information to determine which hypothesis is worth consideration. VisBubbles lets the users load datasets of interest by composing the data (item 7 left) and the good default visualizations (item 7 right) by direct dragging. A new bubble pops up to show a visualization, e.g., as a mesh view (Fig. 1 (2)) or plots. By doing so, the interface supports presentation of information of heterogeneous types. VisBubbles extends the metaphor of bubbles [Bragdon et al. 2010] in that each bubble represents a function unit that is valid for programming and can be interpreted to create a visualization. Here each visualization is created by a certain data type, e.g., spatial wing data are shown as a three-dimensional (3D) mesh.

The second decision making stage is a deterministic hypothesis confirmation stage. In this stage, the scientists compare various design choices and making one choice could have an impact on the analytical results. This stage is incorporated into VisBubbles as means to help us explicitly identify the Matlab programming support (Fig. 1 (10)). Any parameters derived from the Matlab can be placed in the authorized data bubble (Fig. 1(12)), which can also be composed with the good default visualizations (Fig. 1(7) right) to generate visualizations (Fig. 1(11)).

The second inspiration is derived from Ware: visualization should "serve as its own memory" so that users can simply glance at the external interface to comprehend and recall information [Ware 2008]. Thus our interface externalizes the data analysis sequence by presenting information to the interface as visualization bubbles to allow multiple-bubble presentation, rather than hiding the query process.

Unlike windows, bubbles do not overlap but instead push each other out of the way, thus supporting simultaneous side-by-side comparison and freeing users from the windows arrangement, so they can focus on their analytical tasks. Bubbles can be moved freely in space to leverage the action sequence of the data analysis process (considering humans' limited mnemonic abilities of 7 ± 2 items). Bubbles can also be grouped to form a flexible multiple view environments. When grouped, data are automatically linked to support interactive queries in such a way that interacting in one view activates the same interactions in all other grouped views.

5 Conclusion

We have designed a workflow-driven framework that has integrated programming, visualization, and interaction in one framework. Biologists participating in the design suggested that VisBubbles enhances their data query capabilities, enables reuse of their analytical pipelines, and facilitates new student training, previously prohibitively difficult with data intensive computation.

Acknowledgements

This work was supported in part by NSF awards IIS-1018769, IIS-1016623, IOS-0723392, EPS-0903234, and DBI-1062057, and by the Lucas foundation award DE-01316. All opinions, findings, conclusions, or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

BRAGDON, A., REISS, S., ZELEZNIK, R., KARUMURI, S., CHEUNG, W., KAPLAN, J., COLEMAN, C., ADEPUTRA, F., AND LAVIOLA JR, J. 2010. Code bubbles: rethinking the user interface paradigm of integrated development environments. *ACM/IEEE International Conference on Software Engineering*, 455–464.

CALLAHAN, S., FREIRE, J., SANTOS, E., SCHEIDEGGER, C., SILVA, C., AND VO, H. 2006. VisTrails: visualization meets data management. In *ACM SIGMOD International Conference on Management of Data*, ACM, 745–747.

JAMES, W., BOWERS, F., AND SKRUPSKELIS, I. 1975. *The meaning of truth*, vol. 2. Harvard University Press.

PARKER, S., AND JOHNSON, C. 1995. SCIRun: a scientific programming environment for computational steering. In *Supercomputing*, IEEE, 52–52.

UPSON, C., JR., T. F., KAMINS, D., LAIDLAW, D. H., SCHLEGEL, D., VROOM, J., GURWITZ, R., AND VAN DAM, A. 1989. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications 9*, 4 (July), 30–42.

WARE, C. 2008. *Visual thinking for design*. Morgan Kaufmann Pub.