Eurographics Workshop on Visual Computing for Biology and Medicine (2010)
Dirk Bartz, Charl Botha, Joachim Hornegger and Raghu Machiraju (Editors)

Does software engineering pay off for research? Lessons learned from the Caleydo project

M. Streit[1], A. Lex[1], H. Doleisch[2] and D. Schmalstieg[1]

[1]Graz University of Technology, Austria
[2]SimVis GmbH, Austria

Abstract

Caleydo is a visualization research project for biomolecular data analysis which aims to deliver both scientific results as well as software actually usable by domain experts. Based on our experiences we describe lessons learned from a user-centered development process which are potentially valuable for visualization researchers in interdisciplinary projects.

1. Introduction

Many visualization research projects address current problems from various scientific domains and are thus interdisciplinary. Consequently, both researchers from the problem domain as well as from the visualization domain need to learn a lot about the respective other domain. Biomedicine is one such field where interdisciplinary research is very fruitful but also very time-consuming. The visualization experts have to learn the user's language and try to understand their problems to come up with solutions. On the other side, domain experts have to invest time and need to be open minded for innovative and novel applications. Van Wijk [vW06] identified user-centered design as the royal road for a successful interdisciplinary visualization research project.

Over the last four years we have developed Caleydo, a visualization framework for the analysis of biomolecular data. We employ a user-centered design approach and aim to deliver usable software to end-users but also to be a platform for new visualization research ideas.

2. Caleydo Framework: An Overview

Caleydo [SLK*09, LSKS10] focuses on visualizing biomolecular data, for example gene expression data, and bringing it into context with meta-data about genes, proteins or cellular processes. While the biomolecular data is provided by the user, containing for example snapshots of several patient's gene expression, the meta-information (especially pathway graphs which depict cellular processes) are either included in the software or loaded from the web on demand. Figure 1 shows a sample analysis session.

Caleydo is available free of charge for academic use from `http://www.caleydo.org` and runs under Windows and Linux. Basic building blocks of the framework are multiple-coordinated views, linking & brushing as well as other established techniques and paradigms. The core system can be extended with visual as well as analytical interfaces via a plug-in mechanism. Available visual interfaces are for instance parallel coordinates, heat maps and scatterplots. Caleydo also includes novel visualization approaches, such as the Bucket [LSKS10] which allows to manage up to 20 two-dimensional views and shows the relations between these using visual links (see Figure 1). The Bucket is based on style sheets which make it possible to arbitrarily arrange views in a 3D scene and thus facilitate rapid prototyping for view arrangement research. Included analytical interfaces are the R statistics toolkit and WEKA which, for instance, provide clustering algorithms.

3. Lessons Learned

Developing usable software for end-users is contradictory to paper-driven research where ideas are realized as prototypes. For a research project that tries to satisfy both, this balancing act is difficult. Success of visualization projects in academia is measured by the scientific output in terms of papers, suggesting that solely creating research prototypes is sufficient. However, for long-term, high-quality research, this approach can be hindering, as usable applications can serve as the

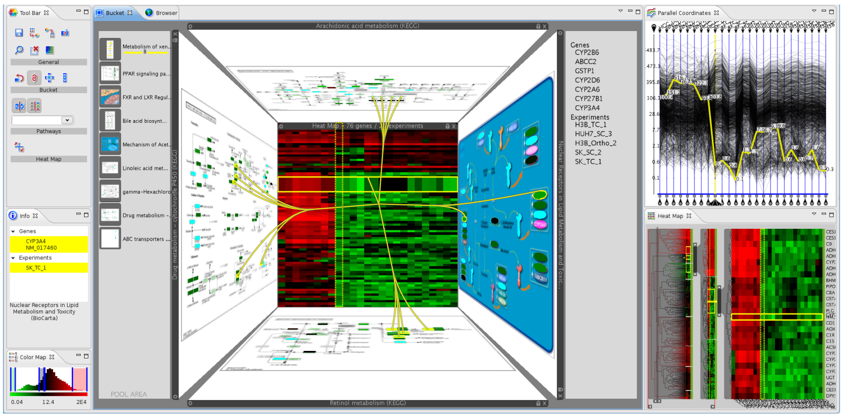

Figure 1: *Screenshot of the Caleydo framework [LSKS10] showing pathways and gene expression data in the Bucket setup. Visual links emphasize interdependencies between related pieces of information.*

basis for new and interesting requirements from end-users leading to new research questions. In addition, complex, interdisciplinary research questions often cannot be answered with a single prototype application built from scratch. Therefore, we believe that spending time to carefully design a reusable framework pays off in the long run. In the following we want to give advice on some issues that arose during the project. We believe that following those points, or at least considering their consequences, can benefit many research-centered software projects:

- **Tell your users what to expect.** Despite efforts by software-engineering-affine researchers, academic software will never reach product level – which should be made explicit to the domain partners from the beginning. However, the collaborators can expect new approaches for challenging problems which they can publish in their own community as well.
- **Flexible plug-in structure.** Dividing the code base into plug-ins allows to create a core system with basic features such as data management, parsing, event systems, etc. Research prototypes of new visualization techniques as well as student projects can then be developed in a separate plug-in – a sandbox environment. This helps to avoid side-effects.
- **Maintain a stable code base.** Paper-driven software development under time-pressure will always lead to an unstable framework state. However, it is essential to regularly reserve time for refactoring, testing, clean-up and documentation. This makes it possible to include new features in a stable release and creates a solid base for future endeavors.
- **Bugs and software problems will occur - minimize their impact!** To alleviate user's frustration when errors occur, measures such as auto-saving analysis sessions and thereby enabling fast recovery, should be taken.
- **Open your code base for research partners.** While a full open-source strategy is controversial especially when future commercialization is possible, we made good experiences with giving research partners access to source-code. This has many positive effects: They can use a feature-rich framework enabling them to realize innovative ideas faster, which can lead to joint publications and a better visibility in the community.

4. Conclusion

In summary, a trade-off between spending time on non-scientific framework development and "real" research work is a practicable way for an interdisciplinary project. While this poster cannot give a universally valid answer to the question raised in the title, it aims to inspire discussions within the community and particularly at the VCBM workshop. However, from our perspective the answer to the question is a definite **yes**.

5. Acknowledgements

The authors want to thank their partners from the Institute of Pathology at the Medical University of Graz for the fruitful collaboration over the years. This work was funded by the Austrian Research Promotion Agency (FFG) through the InGeneious project (385567).

References

[LSKS10] Lex A., Streit M., Kruijff E., Schmalstieg D.: Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *2010 IEEE Pacific Visualization Symposium (PacificVis)* (Taipei, Taiwan, 2010), pp. 57–64.

[SLK*09] Streit M., Lex A., Kalkusch M., Zatloukal K., Schmalstieg D.: Caleydo: Connecting pathways and gene expression. *Bioinformatics 25*, 20 (July 2009), 2760–2761.

[vW06] van Wijk J. J.: Bridging the gaps. *IEEE Computer Graphics and Applications 26*, 6 (2006), 6–9.