

# Visualizing Software Architecture Comparison of a Web-based Financial Application in ExplorViz

Master's Thesis

Josefine Wegert

May 10, 2018

KIEL UNIVERSITY  
DEPARTMENT OF COMPUTER SCIENCE  
SOFTWARE ENGINEERING GROUP

Advised by: Prof. Dr. Wilhelm Hasselbring  
M.Sc. Christian Zirkelbach  
Dr. Achim Held, PPI AG



### **Statutory Declaration**

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Kiel,

---



# Abstract

Spotting an unknown number of changes between two visualizations of a complex software application is a challenging task and error-prone without tool support. However, people are interested in spotting changes in software applications at one glance. They want to track changes in situations where the application faces varied loads or during the evolution of the software application.

ExplorViz visualizes the behavior of software systems based on runtime data. These visualizations support program comprehension. However, currently it lacks the ability to handle two models and the ability to compare two models of a software system. In this thesis, we upgrade ExplorViz to the effect that it is able to merge two models of a software application. For that reason, we extend ExplorViz by an integrated software architecture comparison view that additionally provides highlighting of model elements based on their state. This enables the user to see at a glance which elements have been added, deleted, modified or remained unchanged between the two models.

Subsequently, we apply our software architecture comparison view to the web-based financial application LB-Rating and furthermore, conduct a qualitative usability study with professional software engineers. In this study the participants are asked to solve tasks and to comment on the usability. The study reveals that minor changes in providing feedback to the user will help to ease solving tasks that aim at comparing two models.

# Bibliography

- [Albert and Tullis 2013] W. Albert and T. Tullis. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013. (Cited on page 10)
- [Baum et al. 2017] D. Baum, J. Schilbach, P. Kovacs, U. Eisenecker, and R. Muller. GETAVIZ: generating structural, behavioral, and evolutionary views of software systems for empirical evaluation. In: *IEEE Working Conference on Software Visualization (VISSOFT)*. 2017. (Cited on page 70)
- [Bennett and Rajlich 2000] K. H. Bennett and V. T. Rajlich. Software maintenance and evolution. In: *Proceedings of the conference on The future of Software engineering - ICSE '00*. 2000. (Cited on pages 5, 6)
- [Diehl 2007] S. Diehl. *Software visualization: visualizing the structure, behaviour, and evolution of software*. Springer-Verlag New York, Inc., 2007. (Cited on page 7)
- [Eichhorst 2017] F. Eichhorst. Analyse der microservices eines digitalen marktplatzes mittels explorviz. Masterarbeit. Kiel University, Oktober 2017. URL: <http://eprints.uni-kiel.de/39982/>. (Cited on page 24)
- [Fielding and Taylor 2000] R. T. Fielding and R. N. Taylor. *Architectural styles and the design of network-based software architectures*. Volume 7. University of California, Irvine Doctoral dissertation, 2000. (Cited on page 13)
- [Fittkau et al. 2015a] F. Fittkau, A. Krause, and W. Hasselbring. Exploring software cities in virtual reality. In: *IEEE 3rd Working Conference on Software Visualization (VISSOFT 2015)*. IEEE, Sept. 2015. URL: <http://eprints.uni-kiel.de/29388/>. (Cited on page 9)
- [Fittkau et al. 2017] F. Fittkau, A. Krause, and W. Hasselbring. Software landscape and application visualization for system comprehension with ExplorViz. *Information and Software Technology* (2017). DOI: 10.1016/j.infsof.2016.07.004. (Cited on pages 2, 8, and 11)
- [Fittkau et al. 2015b] F. Fittkau, S. Roth, and W. Hasselbring. Explorviz: visual runtime behavior analysis of enterprise application landscapes. In: *23rd European Conference on Information Systems (ECIS 2015)*. Mai 2015. URL: <http://eprints.uni-kiel.de/28067/>. (Cited on page 11)
- [Fittkau et al. 2014] F. Fittkau, P. Stelzer, and W. Hasselbring. Live visualization of large software landscapes for ensuring architecture conformance. In: *ECSAW 2nd International Workshop on Software Engineering for Systems-of-Systems 2014 (SESoS 2014)*. ACM, Aug. 2014. URL: <http://eprints.uni-kiel.de/25290/>. (Cited on page 71)
- [Harry M. Sneed 2013] R. S. Harry M. Sneed. *Softwareevolution*. Dpunkt.Verlag GmbH, Sept. 11, 2013. URL: [http://www.ebook.de/de/product/20003436/harry\\_m\\_sneed\\_richard\\_seidl\\_softwareevolution.html](http://www.ebook.de/de/product/20003436/harry_m_sneed_richard_seidl_softwareevolution.html). (Cited on pages 5, 6)

## Bibliography

- [Heffelfinger 2017] D. R. Heffelfinger. *Java ee 8 application development: develop enterprise applications using the latest versions of cdi, jax-rs, json-b, jpa, security, and more*. Packt Publishing - ebooks Account, 2017. URL: <https://www.amazon.com/Java-Application-Development-Enterprise-applications/dp/1788293673?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xml2&camp=2025&creative=165953&creativeASIN=1788293673>. (Cited on page 13)
- [Khan et al. 2013] T. Khan, H. Barthel, A. Ebert, and P. Liggesmeyer. Ecity: a tool to track software structural changes using an evolving city. In: *29th IEEE International Conference on Software Maintenance (ICSM)*. 2013. (Cited on page 70)
- [Knight and Munro 2000] C. Knight and M. Munro. Virtual but visible software. In: *IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics*. 2000. (Cited on page 12)
- [Lehman and Belady 1985] M. M. Lehman and L. A. Belady. *Program evolution: processes of software change*. Academic Press Professional, Inc., 1985. (Cited on page 6)
- [Lehman and Ramil 2003] M. M. Lehman and J. F. Ramil. Software evolution—background, theory, practice. *Information Processing Letters* (2003). (Cited on page 6)
- [Lehman 1980] M. Lehman. Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE* (1980). (Cited on page 6)
- [Likert 1932] R. Likert. A technique for the measurement of attitudes. *Archives of psychology* (1932). (Cited on page 56)
- [Nielsen 2012] J. Nielsen. How many test users in a usability study. *Nielsen Norman Group* 4.06 (2012). (Cited on pages 56 and 67)
- [Novais et al. 2013a] R. L. Novais, C. Nunes, A. Garcia, and M. Mendonca. Sourceminer evolution: a tool for supporting feature evolution comprehension. In: *29th IEEE International Conference on Software Maintenance (ICSM)*. 2013. (Cited on page 69)
- [Novais et al. 2013b] R. L. Novais, A. Torres, T. S. Mendes, M. Mendonça, and N. Zazworka. Software evolution visualization: a systematic mapping study. *Information and Software Technology* (2013). (Cited on pages 6, 8, 9)
- [Sommerville 2015] I. Sommerville. *Software engineering, global edition*. 10th edition. Pearson Education Limited, July 29, 2015. URL: [http://www.ebook.de/de/product/25939962/ian\\_sommerville\\_software\\_engineering\\_global\\_edition.html](http://www.ebook.de/de/product/25939962/ian_sommerville_software_engineering_global_edition.html). (Cited on page 7)

## Bibliography

- [Van Hoorn et al. 2012] A. van Hoorn, J. Waller, and W. Hasselbring. Kieker: A framework for application performance monitoring and dynamic software analysis. In: *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering (ICPE 2012)*. 2012. (Cited on page 20)
- [Wettel 2010] R. Wettel. Software systems as cities. PhD thesis. Università della Svizzera Italiana, 2010. (Cited on page 71)
- [Zirkelbach 2017] C. Zirkelbach. Juggling with data: on the lack of database monitoring in long-living software systems. In: *4th Collaborative Workshop on Evolution and Maintenance of Long-Living Software Systems (EMLS)*. 2017. (Cited on page 2)
- [Zirkelbach et al. 2015] C. Zirkelbach, W. Hasselbring, and L. Carr. Combining kieker with gephi for performance analysis and interactive trace visualization. In: *Symposium on Software Performance 2015: Joint Developer and Community Meeting of Descartes/Kieker/Palladio*. 2015. (Cited on page 2)