Software Impacts 6 (2020) 100034

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

ExplorViz: Research on software visualization, comprehension and collaboration

Wilhelm Hasselbring *, Alexander Krause, Christian Zirkelbach

Software Engineering Group, Kiel University, D-24098 Kiel, Germany

ARTICLE INFO

Keywords:
Software engineering research
Software visualization
Software comprehension
Software collaboration
Virtual reality

ABSTRACT

ExplorViz supports research on software visualization, software comprehension tasks and software collaboration. To achieve this, ExplorViz provides multi-level visualization from the software landscape layer toward the level of individual software applications. Via immersive 3D visualizations in virtual reality, ExplorViz also supports collaboration in software development teams. The research tool development commenced in 2012, and grew toward a powerful, extensible open-source software that has been employed in a variety of software engineering research projects. In this paper, we review its history, development and research impact.

Code metadata

Current code version	1.5.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2020-36
Permanent link to Reproducible Capsule	
Legal Code License	Apache License, Version 2.0
Code versioning system used	Git
Software code languages, tools, and services used	Java, JavaScript, TypeScript
Compilation requirements, operating environments & dependencies	Java 11 or higher, Docker
If available Link to developer documentation/manual	https://github.com/ExplorViz/Docs/wiki
Support email for questions	https://www.explorviz.net/team.php

Software metadata

Current software version	1.5.0
Permanent link to executables of this version	https://www.explorviz.net/download.php
Permanent link to Reproducible Capsule	
Legal Software License	Apache License, Version 2.0
Computing platforms/Operating Systems	Various, including Linux, OS X, Microsoft Windows, Unix-like
Installation requirements & dependencies	Docker
Link to user manual	https://github.com/ExplorViz/docker-configuration
Support email for questions	https://www.explorviz.net/team.php

1. Software visualization, comprehension and collaboration

ExplorViz uses dynamic analysis techniques to provide live trace visualization of the communication in large software landscapes. It targets software system and program comprehension in those landscapes while still providing details on the communication within an application. A landscape perspective enriches previous system visualizations with additional abstraction levels for efficient comprehension of communication and interaction among software applications. On the software application level perspective, ExplorViz utilizes the 3D city metaphor [1] combined with an interactive concept of showing details that are in focus of the analysis. For best accessibility, ExplorViz is implemented as a web-based tool. With the 3D city metaphor visualization of a software application, new collaborative interaction styles have been investigated with ExplorViz for a more effective and efficient program comprehension process in software development teams. The usability and effectiveness of ExplorViz have been investigated in

* Corresponding author.
E-mail addresses: hasselbring@email.uni-kiel.de (W. Hasselbring), akr@informatik.uni-kiel.de (A. Krause), czi@informatik.uni-kiel.de (C. Zirkelbach).

https://doi.org/10.1016/j.simpa.2020.100034
Received 20 August 2020; Accepted 1 September 2020

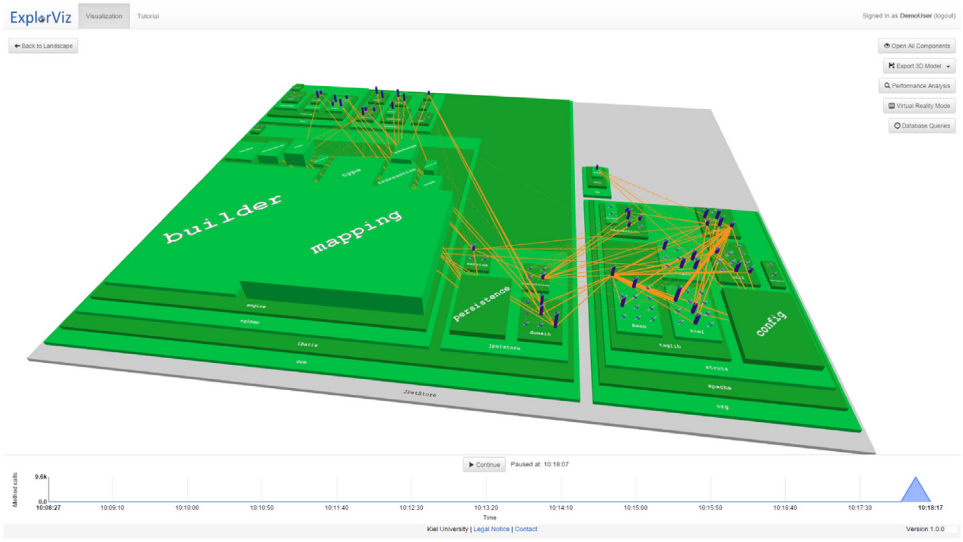

Fig. 1. ExplorViz utilizes the 3D city metaphor for visualization on the software application level perspective.

various empirical experiments which resulted in increased efficiency and effectiveness over competing approaches.

The core research topics in ExplorViz are *Software Visualization*, *Software Comprehension* and *Collaboration in Software Engineering*, as well as the combination of those topics:

Software Visualization is concerned with the visualization of information related to software systems, such as the architecture, its runtime behavior, and its development process by means of static, interactive or animated 2D or 3D visual representations of their structure, execution, behavior, and evolution. As an example, Fig. 1 shows how ExplorViz utilizes the 3D city metaphor for visualizing the application level perspective of software systems.

Software Comprehension is concerned with the ways software engineers maintain existing software. Software comprehension is an important cognitive process in software development, because developers spend most of their time with understanding existing software.

Collaboration in Software Engineering is concerned with coordinating the work of software engineers to design, implement, and maintain large software systems in teams. ExplorViz supports collaboration in software comprehension and software design.

2. Development history

The open source research project ExplorViz [2–4] started in 2012 as part of a PhD thesis [5] and is further developed and actively maintained until today. ExplorViz enables a live monitoring and visualization of large software landscapes. Therefore, the tool offers two types of visualizations — a landscape-level and an application-level perspective. The first provides an overview of a monitored software landscape consisting of several servers, applications, and communication in-between. The second perspective visualizes a single application within the software landscape and reveals its underlying architecture. The tool has the objective to aid the process of system and program comprehension for developers and operators.

The software is developed from the beginning on GitHub with some core developers and many collaborators over the time (more than 40 students). Several extensions have been implemented since the first version, which enhanced the tool's feature set.

In 2016 we started to migrate and modernize our employed architecture and technology stack to improve the maintainability of the software and the collaboration among the developers. This way, ExplorViz became a research object itself, as study object for software modularization and modernization.

With the new modular microservices architecture, new project developers, e.g., students, do not have to understand the complete project from the beginning. They can now extend the core by implementing new mechanics on the basis of plug-in extensions. With this modernization, the previous own, specialized monitoring component has been replaced by the generic Kieker monitoring framework [6,7]. Fig. 2 shows the new modularized, extensible software architecture of ExplorViz [8].

Since 2017, we continuously extended and improved the virtual reality (VR) features of ExplorViz. Starting with a single-user VR approach as an alternative to existing application-level visualizations of ExplorViz on the screen, we realized a first multi-user VR approach in 2019, and achieved a fully collaborative multi-user VR approach offering landscape-level perspective in 2020. The latest version allows multiple users with head-mounted displays and specific controllers to collaboratively explore and comprehend monitored software systems in VR. Based on several gestures mapped to related controllers the users are able to interact with the immersive software visualization and thus can perform system and program comprehension tasks in teams. Even developers located at different sites are now able to collaboratively utilize ExplorViz.

3. Research impact

Research areas where ExplorViz' visualization was successfully employed for software engineering research include software comprehension [9,10], live visualization [11–14], architecture conformance checking [15], performance analysis [16], 3D printing [17] and virtual reality [18,19]. Related to ExplorViz, SynchroVis [20] emerged as a tool for analyzing concurrency via 3D visualization.

Besides employing ExplorViz in research on software visualization, comprehension and collaboration, ExplorViz has also been used as a research object itself. This includes scalable and elastic processing of large volumes of monitoring data [21], application discovery [22] and migrating monolithic software systems toward microservice architectures [8,23–25].

4. Industry impact

ExplorViz was also employed in several industrial collaborations. Examples are PPI AG (Kiel) and Adesso SE (Hamburg) [26]. These industrial collaborations and case studies also serve as evaluation of

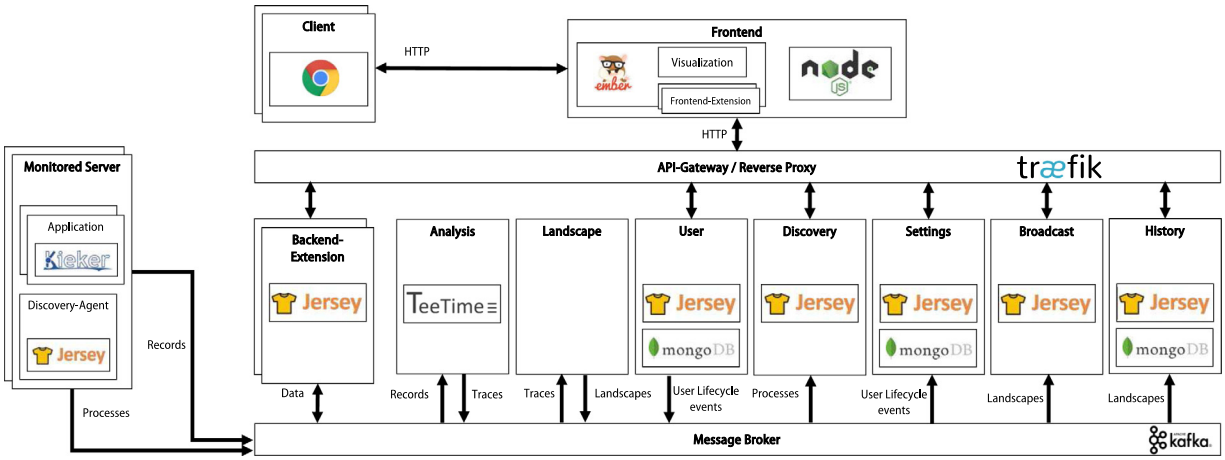

Fig. 2. The modularized, extensible software architecture of ExplorViz [8].

the ExplorViz approach, influence the development of ExplorViz and provide access to real-world data.

ExplorViz is open source research software [27] licensed under the Apache License, Version 2.0, such that it may be utilized commercially without any restrictions. Such a license is a good legal framework for technology transfer. The "business model" is not based on envisioned revenue via licensing, instead we follow an open source business model based on *impact* of the software. More frequent use of the software means more impact, in this case. Such impact is a great foundation for follow-up projects. We also use ExplorViz as example software system for software engineering education.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Many colleagues contributed to ExplorViz in different ways and intensities.[1] Note that we do not only acknowledge contributions to source code. The contributing researchers are usually involved because they are working on a ExplorViz-related research project. Students usually contribute to ExplorViz as part of their work on ExplorViz-related study theses or their employment as student assistants.

References

[1] R. Wettel, M. Lanza, Visualizing software systems as cities, in: Proceedings of the 4th IEEE International Workshop on Visualizing Software for UnderstandIng and Analysis, 2007, pp. 92–99, http://dx.doi.org/10.1109/VISSOF.2007.4290706.

[2] F. Fittkau, J. Waller, C. Wulf, W. Hasselbring, Live trace visualization for comprehending large software landscapes: The ExplorViz approach, in: Proceedings of the IEEE International Working Conference on Software Visualization (VISSOFT 2013), 2013, pp. 1–4, http://dx.doi.org/10.1109/VISSOFT.2013.6650536.

[3] F. Fittkau, S. Roth, W. Hasselbring, ExplorViz: visual runtime behavior analysis of enterprise application landscapes, in: Proceedings of the European Conference on Information Systems (ECIS 2015 Completed Research Papers), AIS Electronic Library, 2015, pp. 1–13, http://dx.doi.org/10.18151/7217313.

[4] F. Fittkau, A. Krause, W. Hasselbring, Software landscape and application visualization for system comprehension with ExplorViz, Inf. Softw. Technol. 87 (2017) 259–277, http://dx.doi.org/10.1016/j.infsof.2016.07.004.

[5] F. Fittkau, Live Trace Visualization for System and Program Comprehension in Large Software Landscapes (Ph.D. thesis), Kiel University, 2015.

[6] W. Hasselbring, A. van Hoorn, Kieker: A monitoring framework for software engineering research, Softw. Impacts 5 (2020) 1–5, http://dx.doi.org/10.1016/j.simpa.2020.100019.

[7] A. van Hoorn, J. Waller, 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 '12), ACM, 2012, pp. 247–248, http://dx.doi.org/10.1145/2188286.2188326.

[8] C. Zirkelbach, A. Krause, W. Hasselbring, The collaborative modularization and reengineering approach CORAL for open source research software, Int. J. Adv. Softw. 13 (1 & 2) (2020) 39–49, URL http://www.iariajournals.org/software/tocv13n12.html.

[9] F. Fittkau, A. Krause, W. Hasselbring, Hierarchical software landscape visualization for system comprehension: A controlled experiment, in: Proceedings of the 3rd IEEE Working Conference on Software Visualization (VISSOFT 2015), IEEE, 2015, pp. 36–45, http://dx.doi.org/10.1109/VISSOFT.2015.7332413.

[10] F. Fittkau, S. Finke, W. Hasselbring, J. Waller, Comparing trace visualizations for program comprehension through controlled experiments, in: Proceedings of the IEEE International Conference on Program Comprehension (ICPC 2015), IEEE, 2015, pp. 266–276, http://dx.doi.org/10.1109/ICPC.2015.37.

[11] F. Fittkau, A. van Hoorn, W. Hasselbring, Towards a dependability control center for large software landscapes, in: Tenth European Dependable Computing Conference (EDCC 2014), IEEE, 2014, pp. 58–61, http://dx.doi.org/10.1109/EDCC.2014.12.

[12] R. Heinrich, R. Jung, C. Zirkelbach, W. Hasselbring, R. Reussner, An architectural model-based approach to quality-aware devops in cloud applications, in: I. Mistrik, R. Bahsoon, N. Ali, M. Heisel, B. Maxim (Eds.), Software Architecture for Big Data and the Cloud, Elsevier, 2017, pp. 69–89, http://dx.doi.org/10.1016/B978-0-12-805467-3.00005-3.

[13] R. Heinrich, C. Zirkelbach, R. Jung, Architectural runtime modeling and visualization for quality-aware devops in cloud applications, in: Proceedings of the IEEE International Conference on Software Architecture Workshops (ICSAW 2017), 2017, pp. 199–201, http://dx.doi.org/10.1109/ICSAW.2017.33.

[14] W. Hasselbring, R. Heinrich, R. Jung, A. Metzger, K. Pohl, R. Reussner, E. Schmieders, IObserve: Integrated Observation and Modeling Techniques to Support Adaptation and Evolution of Software Systems, Technical Report TR-1309, Kiel University, Kiel, Germany, 2013.

[15] F. Fittkau, P. Stelzer, W. Hasselbring, Live visualization of large software landscapes for ensuring architecture conformance, in: Proceedings of the 2014 European Conference on Software Architecture Workshops (ECSAW 2014), ACM, 2014, pp. 28:1–28:4, http://dx.doi.org/10.1145/2642803.2642831.

[16] C. Zirkelbach, W. Hasselbring, F. Fittkau, L. Carr, Performance Analysis of Legacy Perl Software Via Batch and Interactive Trace Visualization, Technical Report TR 1509, Kiel University, 2015.

[17] F. Fittkau, E. Koppenhagen, W. Hasselbring, Research perspective on supporting software engineering via physical 3D models, in: Proceedings of the 3rd IEEE International Working Conference on Software Visualization (VISSOFT 2015), IEEE, 2015, pp. 125–129, http://dx.doi.org/10.1109/VISSOFT.2015.7332422.

[18] F. Fittkau, A. Krause, W. Hasselbring, Exploring software cities in virtual reality, in: Proceedings of the 3rd IEEE International Working Conference on Software Visualization (VISSOFT 2015), IEEE, 2015, pp. 130–134, http://dx.doi.org/10.1109/VISSOFT.2015.7332423.

[19] C. Zirkelbach, A. Krause, W. Hasselbring, Hands-on: Experiencing Software Architecture in Virtual Reality, Technical Report TR 1809, Kiel University, 2019.

[1] A list of ExplorViz contributors is available at https://www.explorviz.net/team.php.

[20] J. Waller, C. Wulf, F. Fittkau, P. Döhring, W. Hasselbring, SynchroVis: 3D visualization of monitoring traces in the city metaphor for analyzing concurrency, in: 1st IEEE International Working Conference on Software Visualization (VISSOFT 2013), 2013, pp. 1–4, http://dx.doi.org/10.1109/VISSOFT.2013.6650520.

[21] F. Fittkau, W. Hasselbring, Elastic application-level monitoring for large software landscapes in the cloud, in: M. Villari S. Dustdar (Ed.), Service Oriented and Cloud Computing, in: Lecture Notes in Computer Science, vol. 9306, Springer-Verlag, 2015, pp. 80–94, http://dx.doi.org/10.1007/978-3-319-24072-5_6.

[22] A. Krause, C. Zirkelbach, W. Hasselbring, Simplifying software system monitoring through application discovery with ExplorViz, Softwaretechnik-Trends 39 (3) (2019) 46–48, (Proceedings of the 9th Symposium on Software Performance (SSP 2019)). URL http://pi.informatik.uni-siegen.de/stt/39_3/index.html.

[23] C. Zirkelbach, A. Krause, W. Hasselbring, Modularization of research software for collaborative open source development, in: The Ninth International Conference on Advanced Collaborative Networks, Systems and Applications (COLLA 2019), 2019, pp. 1–7.

[24] C. Zirkelbach, A. Krause, W. Hasselbring, On the Modularization of ExplorViz Towards Collaborative Open Source Development, Technical Report TR 1902, Kiel University, 2019.

[25] C. Zirkelbach, A. Krause, W. Hasselbring, On the modernization of ExplorViz towards a microservice architecture, in: 5th Collaborative Workshop on Evolution and Maintenance of Long-Living Software Systems (EMLS 2018), Vol-2066, CEUR Workshop Proceedings, Ulm, Germany, 2018, pp. 39–42, URL http://ceur-ws.org/Vol-2066/.

[26] A. Krause, C. Zirkelbach, W. Hasselbring, S. Lenga, D. Kröger, Microservice decomposition via static and dynamic analysis of the monolith, in: Proceedings of the IEEE International Conference on Software Architecture Companion (ICSA-C), 2020, pp. 9–16, http://dx.doi.org/10.1109/ICSA-C50368.2020.00011.

[27] W. Hasselbring, L. Carr, S. Hettrick, H. Packer, T. Tiropanis, Open source research software, Computer 53 (8) (2020) 84–88, http://dx.doi.org/10.1109/MC.2020.2998235.