Guest editorial of the special section on software visualization

Alexandre Bergel

Pleiad, University of Chile

Fabian Beck

VISUS, University of Stuttgart

Software visualization aims at making understandable the inherently complex and abstract data of software systems. Focusing on the code and structure, network diagrams could visualize the architecture of a software system, a timeline might show evolution of the code, or charts potentially reveal runtime metrics of a system. But software is not limited to code and its execution, also the environment that produces and uses it is important to understand and improve development and usage of the software. For example, social network visualizations could help analyze the structure of development teams or the visual analysis of log files might give insights into real usage scenarios of the software. In contrast to statistical analysis or data mining, visualization supports more explorative analysis scenarios and is usually easier to understand and interpret by users such as developers. However, algorithmic solutions and visualization could also blend together in visual analytics systems.

This special section features three distinguished papers on software visualization selected for extension from the Proceedings of the third IEEE Working Conference on Software Visualization (VISSOFT 2015). The conference is a merger of the IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT) and the ACM Symposium on Software Visualization (SOFTVIS). Based on the extensive reviews by the conferences program committee members, the program co-chairs selected and invited for

extension the three best contributions among the twelve technical papers of the conference. All invited authors accepted the invitation and submitted a revised version of their work including a substantial extension of at least 30% new material. Each of the papers was reviewed by two software visualization experts and went through an iterative revision process. Finally, all three invited submissions were accepted due to their high-quality contributions and significant extension.

The paper *"Software Landscape and Application Visualization for System Comprehension with ExplorViz"* investigates the design of map-based visualizations of software architecture. In controlled experiments, the authors study the importance of a hierarchical representation of a software system and the difference of a map-based layout to a radial layout. Furthermore, they explore advantages and problems of a 3D-printed and a virtual reality representation of the map-based visualization. The conference version of the paper is accompanied with an evaluated artifact and was awarded as best paper of the conference.

The paper *"Stable and Predictable Voronoi Treemaps for Software Quality Monitoring"* takes into account the hierarchical organization of software systems and focuses on the evolution of this structure over development time. In particular, the authors introduce an improved algorithm for computing hierarchical Voronoi treemaps. These diagrams are able to represent quantities such as software metrics within a hierarchical structure and follow the organic shapes of Voronoi diagrams. The specific goal of the research was to stabilize the layout of the diagram to make changes traceable along the evolution of the software system.

The paper *"Visualizing and Exploring Software Version Control Repositories using Interactive Tag Clouds over Formal Concept Lattices"* shows the revisions of a software systems in a set of connected word cloud representations. Each word cloud represents occurrence frequencies of entities—like revisions, code artifacts, developers, and time spans—from a different perspective. Formal concept analysis forms the basis of implementing a flexible interaction mechanism that supports sophisticated analysis scenarios. The conference version of this paper provides an evaluated artifact as supplemental material.

The featured papers illustrate both the breadth of topics and diversity of methods in software visualization research. As a bridging element, the source code entities of the studied software systems play a key role in all three approaches. They are visualized as nested boxes, Voronoi cells, or keywords in a word cloud. While the software landscape visualization shows details of the software architecture of one software version, the other two approaches study the evolution of the software system across versions. Like often in software visualization, hierarchies and networks are an important form of data representation. Methods applied in the selected papers include performing controlled user studies, engineering efficient geometric algorithms, and designing advanced visual analysis systems. In general, the selected papers provide distinguished examples of the current state of the art in software visualization.