Visualization of Large Software Architectures and their Evolution

By
Taimur Khan (tkhan@informatik.uni-kl.de)
Supervisor: Prof. Dr. Achim Ebert, Prof. Dr. Hans Hagen, Prof. Dr.-Ing. Peter Liggesmeyer

The foremost task in the comprehension of large-scale software system maintenance and evolution is to understand its architecture before examining lower level details. However, visualization methods and metaphors found in mainstream applications with respect to the above-mentioned criterion are not robust enough to provide interactive exploratory visualization of such large systems. Further, there is hardly any work that depicts how software architecture evolves over time visually. It is the goal of this research to address both these key issues.

There are already a number of free and commercial tools that can be found in both academic and industrial research, with the sole purpose of improving software architecture comprehension through the use of visualization. On the one hand, industrial applications are confined to easy to integrate visualization methods and metaphors that lack the sophistication to handle informative large-scale software architecture visualization. While on the other, academic researchers have developed numerous solutions that have not made it to the mainstream. Our aim is to bridge this gap by addressing the following factors, which we believe researchers have overlooked:

- Proposed visualization methods and metaphors should not significantly deviate from the current solutions – analysts should be comfortable with the software architecture representations visually.
- The ability to monitor, visualize, and interact with large-scale software systems in real-time – be able to deal with the scale and complexity of real-world software applications.

Another key aspect that is believed to be overlooked in ‘theoretical research’ is the magnitude of developers involved in the creation of a software product. This naturally occurring phenomenon may at times lead to unintentional side-effects, such as nonconformity between the planned and actual software architecture states. As such, the visual exploration is required to not only perform compliance checking, but also to keep an eye on how the system changes over time, i.e. have the ability to conduct time-base analysis. Current software evolution research focuses on the time-based visualization of either file systems or lines of code instead of software architecture; it is our belief that more efficient time analysis of large-scale systems can be performed by visualizing at the software architecture level of abstraction.

In order to rectify the above-mentioned issues, it is a prerequisite to have visualization methods and metaphors that optimally utilize display space and convey the maximum amount of information to the user. In order to avoid information overload, suitable filters have to be applied to the data. Clustering algorithms may be used to visually encapsulate the inherent structures of software systems design, in order to visually present relations at different levels of abstraction. This filter functionality combined with modern interaction, navigation, and drill-down functionalities would allow the user to conduct analysis of hierarchical structures in software systems more efficiently than what is possible at the moment.

The focus of this research project is to utilize revolutionary ideas in the area of interactive exploratory visualization and to apply these modern techniques in the context of software architecture maintenance and evolution.