Interactive Visual Analysis of Software Structures

Maintaining complex software systems tends to be a costly activity where software engineers spend a significant amount of time trying to understand the system's structure and behavior. As early as the 1980s, operation and maintenance costs were already twice as expensive as the initial development costs incurred. Since then these costs have steadily increased. The focus of this work is to reduce these costs through novel interactive exploratory visualization concepts and to apply these modern techniques in the context of services offered by software quality analysis.

Costs associated with the understanding of software are governed by specific features of the system in terms of different domains, including re-engineering, maintenance, and evolution. These features are reflected in software measurements or inner qualities such as *extensibility, reusability, modifiability, testability, compatability, or adatability*. The presence or absence of these qualities determines how easily a software system can conform or be customized to meet new requirements. Consequently, the need arises to monitor and evaluate the qualitative state of a software system in terms of these qualities. Using metrics-based analysis, production costs and quality defects of the software can be recorded objectively and analyzed. In practice, there exist a number of free and commercial tools that analyze the inner quality of a software system through the use of software metrics. However, most of these tools focus on software data mining and metrics (computational analysis) and only a few support visual analytical reasoning. Typically, computational analysis tools generate data and software visualization tools facilitate the exploration and explanation of this data through static or interactive visual representations. Tools that combine these two approaches focus only on well-known metrics and lack the ability to examine user defined metrics. Further, they are often confined to simple visualization methods and metaphors, including charts, histograms, scatter plots, and node-link diagrams.

The goal of this work is to develop methodologies that combine computational analysis methods together with sophisticated visualization methods and metaphors through an interactive visual analysis approach. This approach promotes an iterative knowledge discovery process through multiple views of the data where analysts select features of interest in one of the views and inspect data items of the select subset in all of the views. On the one hand, we introduce a novel approach for the visual analysis of software measurement data that captures complete facts of the system, employs a flow-based visual paradigm for the specification of software measurement queries, and presents measurement results through integrated software visualizations. This approach facilitates the on-demand computation of desired features and supports interactive knowledge discovery – the analyst can gain more insight into the data through activities that involve: building a mental model of the system; exploring expected and unexpected features and relations; and generating, verifying, or rejecting hypothesis with visual tools. On the other hand, we have also extended existing tools with additional views of the data for the presentation and interactive exploration of system artifacts and their inter-relations.

Contributions of this work have been integrated into two different prototype tools. First evaluations of these tools show that they can indeed improve the understanding of large and complex software systems.