1 Introduction: Software Map

The software map [Bohnet and Döllner 2011] is a visualization technique based on the 2.5D treemap [Bladh et al. 2004] used to depict the structure and characteristics of one revision of a software system. It is configured by map themes, consisting of a mapping of metrics to visual variables (e.g., ground area, height, color, texture). A software map inhibits the following properties:

- Nested nodes and their padding depict the hierarchy.
- One revision is visualized using one map theme.
- A leaf node depicts a measurable software entity.
- The ground area of the leaf node depicts its size.
- Its height represents a mapped metric, defined by the map theme.
- Its color represents a mapped metric, defined by the map theme.

While software maps are useful to visualize one revision using one map theme, the interactive revision exploration is challenging. A software system under development is a tree comparison problem with both changing metrics and values and topology (category 3 or 4, depending on the metrics used [Guerra-Gomez et al. 2013]). To support the use case, the following concepts and characteristics of a software map are missing:

- Depicting multiple revisions using the same map theme.
- Depicting multiple map themes using the same revision.
- Comparison between different depictions of the same software system.

We propose an extension to software maps where multiple instances using different revisions and map themes are arranged into a matrix, constituting a small multiples visualization [van den Elzen and van Wijk 2013].

2 Concept

We build a small multiples visualization of software maps by arranging the software maps in a grid. The x-axis is used for the time-component (i.e., differing revisions) and the y-axis for different map themes. To configure each software map, we differentiate between the base configuration (including, e.g., the dataset, the layouting algorithm and the node padding) that is used for all software maps and a per small multiples configuration that specifies the used revision and map theme. A graphical user interface allows to configure the revision per column and the map theme per row.

Navigation techniques for the virtual 3D scenes are zoom, pan, and rotate, while the virtual cameras are all synchronized. A focus-context technique allows for highlighting one or multiple software maps for direct comparison.

3 Implementation

The prototype is implemented using C++, Q5 and OpenGL while relying on an attributed point cloud (Trapp et al. 2013) as geometry representation on the GPU and multi-frame sampling [Limberger et al. 2016] for the shading. Two rendering pipelines were implemented and evaluated: a multi-pass approach where each small multiple is rendered in its own draw call and a single-pass approach where all small multiples are rendered using one draw call.

4 Interactive Revision Exploration

References


