

# Viewport for Component Diagrams

Lukas Holy and Premek Brada

Department of Computer Science and Engineering, University of West Bohemia,  
Univerzitni 8, Pilsen, Czech Republic

**Abstract.** This paper describes a viewport technique for use in the visualization of large graphs, e.g. UML component diagrams. This technique should help to work with complex diagrams (hundreds or thousands of components) by highlighting details of the important parts of the diagram and their related surroundings without losing the global perspective. To avoid visual clutter it uses clusters of interfaces and components.

## 1 Introduction

Although software components [2] comprise relatively large parts of systems, nowadays applications can easily consist of hundreds or thousands heavily interconnected ones. Thus their UML component diagrams become large graphs which are difficult to explore for humans. The main problem is how to show the whole diagram and provide enough detailed information at the same time. Diagrams displayed at the desired level of detail become too big to provide a sufficient overview and keep orientation; especially difficult is to trace dependencies between distant components. When displaying the whole diagram on standard screens, individual elements are hard to recognize and often there is visual clutter caused by dependency visualization.

Visualization techniques which handle the complexity, such as off-screen rendering [1], can be used instead of the traditional pan&zoom technique. This paper describes a novel approach called viewport which attempts to reconcile the above mentioned contradictory requirements and helps to explore the dependencies among components in an intuitive way.

## 2 Viewport for Component Diagrams

The proposed technique shows the graph (standard UML component diagram) zoomed-out to provide the appropriate overview of the complete architecture, with elements displayed without details. Besides that it shows selected components in detail inside a *viewport area* plus all their relations with other components in the diagram in an interactive border area (see Figure 1). These relations are for each component clustered into two sets: all provided interfaces (displayed as "lollipops") and all required interfaces (displayed as "sockets").

These interfaces are then connected to clustered proxy components, visually represented as rectangles with rounded corners. Each rectangle represents one

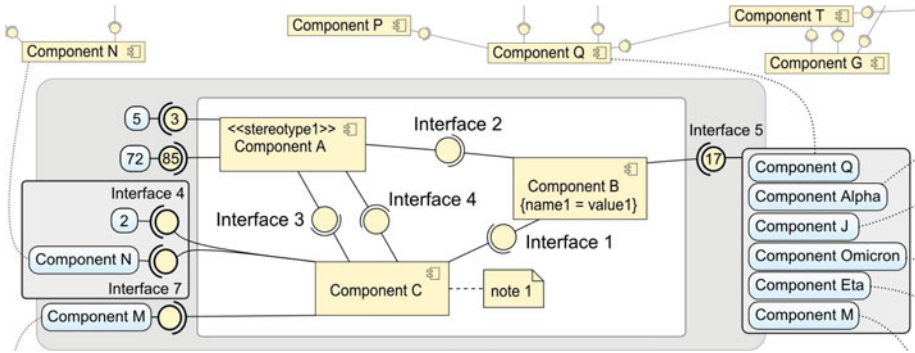


Fig. 1. Viewport for component diagrams

or more components. Numbers inside the clustered interfaces and proxy components represent a desired metric, e.g. the number of elements clustered in a given symbol. One of the key factors of our approach will be the interactivity of the border area, which should comprise user manipulation with clustering of interfaces or components, layout adjustments and selecting the components shown in the viewport.

The viewport technique should enable to explore and understand the dependencies in large diagrams by showing the context of a selected diagram subset. The clustering shall reduce the visual clutter otherwise caused by large number of relations. The proxy elements should reduce the need for the disorienting pan&zoom otherwise necessary while exploring dependencies and provide user relevant information in one place. The viewport can either be placed on a given position in the diagram (there can be more viewports in a diagram) or have a fixed position on the screen.

### 3 Future Work

Important part of the future research on this technique are layout algorithms for components displayed both inside and outside of the viewport. Also, options for automatic suggestion of diagram parts suitable for displaying in viewports based on graph algorithms will be investigated.

### References

1. Frisch, M., Dachselt, R.: Off-screen visualization techniques for class diagrams. In: Proceedings of the 5th International Symposium on Software Visualization, SOFT-VIS 2010, pp. 163–172. ACM, New York (2010)
2. Szyperski, C.: Component Software: Beyond Object-Oriented Programming, 3rd edn. Addison-Wesley / ACM Press (2002)