Hierarchy based Visualization of Large Software Structures

Michael Balzer and Oliver Deussen – University of Konstanz, Germany

Abstract

Modern object-oriented programs are hierarchical systems with many thousands of interrelated subsystems. Visualization helps developers to better comprehend these large and complex systems. This work presents a three-dimensional visualization technique that represents the static structure of object-oriented software using distributions of three-dimensional objects on a two-dimensional plane. The visual complexity is reduced by adjusting the transparency of object surfaces to the distance of the viewpoint. An approach called Hierarchical Net is proposed for a clear representation of the relationships between the subsystems.
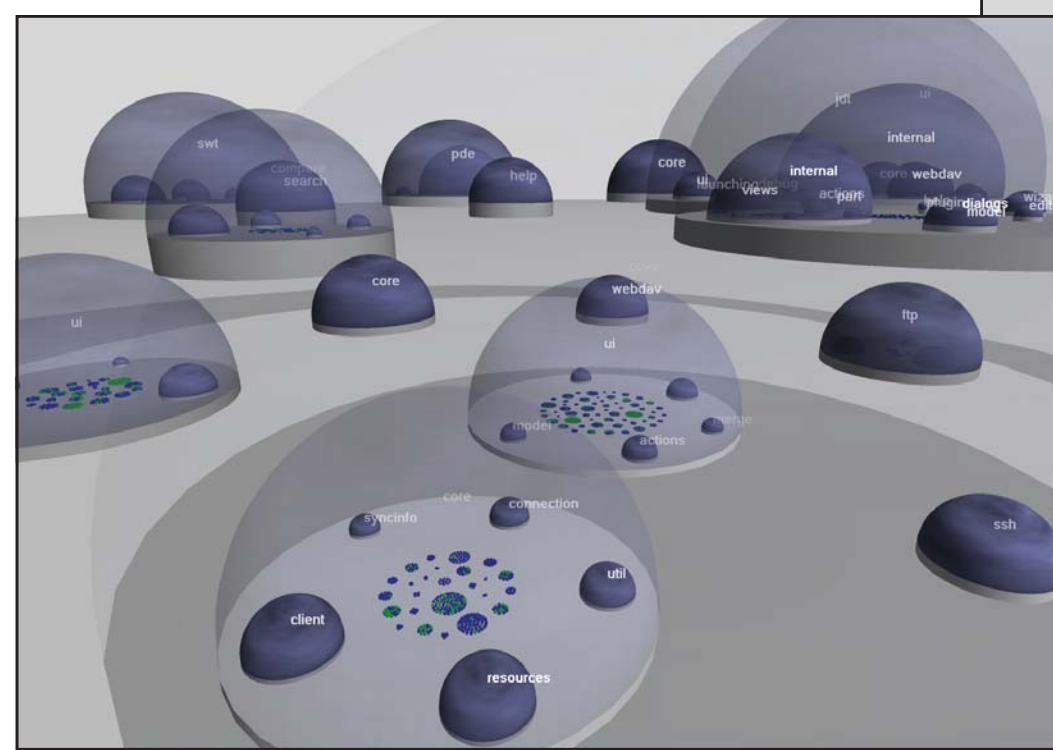
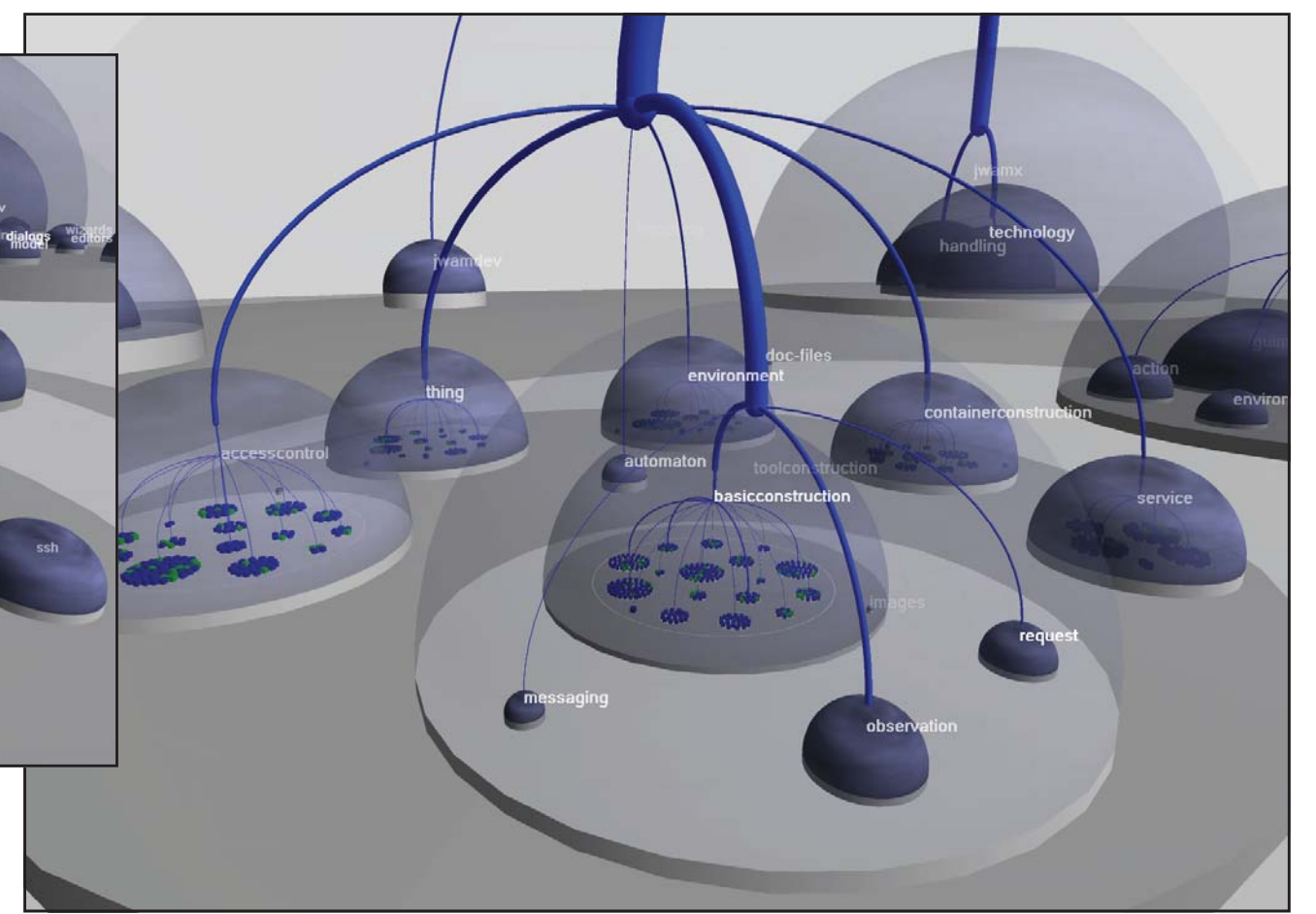Figure 1: Hierarchy based 3D representation of the 'Eclipse' software system

Figure 2: Visualization of entity relations using a Hierarchical Net

1 A Strucural Model of Object-Oriented Software

The models are nested directed graphs:
- nodes correspond to software entities
- inclusion tree corresponds to containment hierarchy
- edges correspond to relations between entities

The models are automatically extracted from the source code of existing software and stored in Rigi Standard Format.
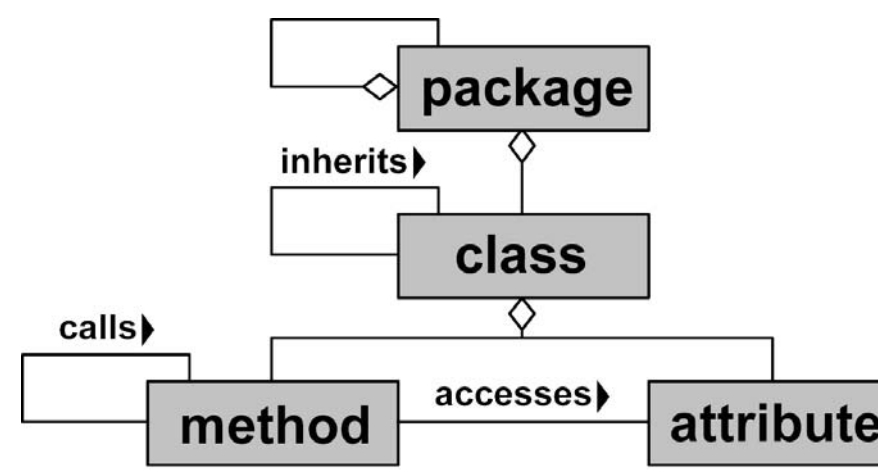

Figure 3: Schema for object-oriented software

System	Entities	Relations	LOC
Eclipse 2.02	112 613	339 161	181 270
JWAM 1.6	11 097	25 081	167 178
JDK 1.4.0 AWT	9 470	20 029	141 267
System X	8 042	19 378	78 220
CrocoCosmo	1 269	2 574	16 832

Table 1: Examples of visualized software systems

2 Hierarchy Base Layout of Entities

The automatic generation of 2D object layouts is based on the component hierarchy of the software system.

First stage: representing the (potentially infinitely deep) package hierarchy by nested hemispheres on a 2D plane

Second stage: representing classes of the packages as circles in the center of the hemispheres

Third stage: representing methods and attributes within the classes represented by circles

The size of packages and classes ist adjusted to the number of contained methods and attributes.

As a tool for the generation of this distributions we use the relaxation of Voronoi diagrams.
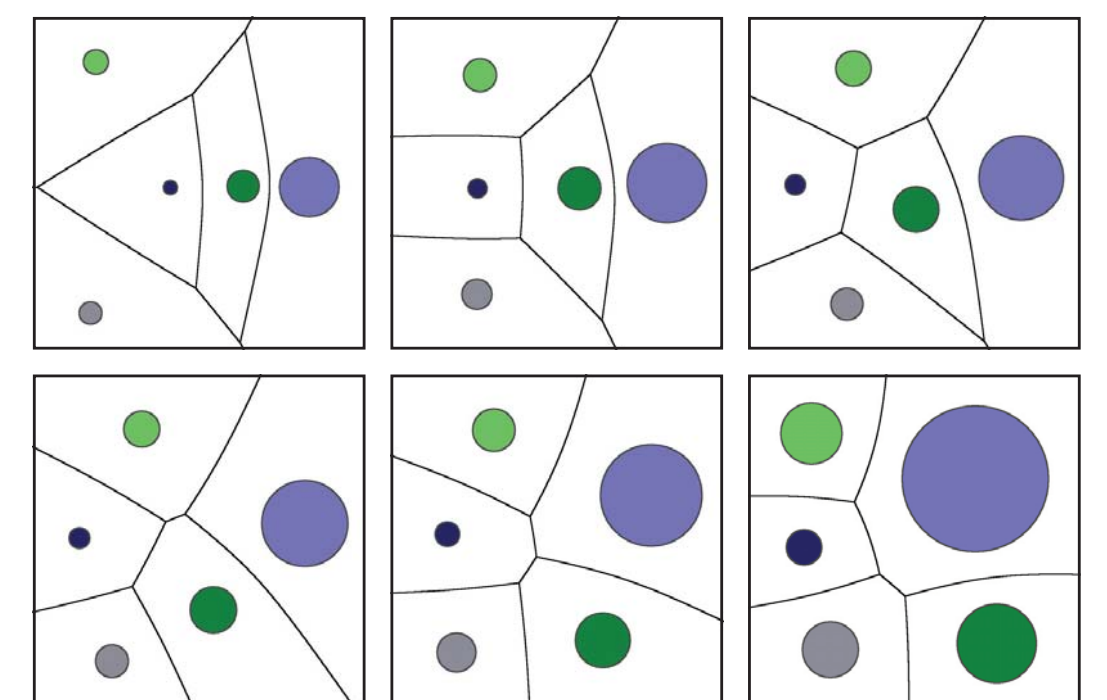

Figure 4: Relaxation of Voronoi diagrams under the criterion of Maximum Growth

3 Hierarchical Nets

The relations between the software entities are an important part of software structures.

Direct line connections between entities would result in very unclear representations.

A solution are Hierarchical Nets, whereby the relations are routed according to the hierarchy levels of the software entities.

Relations of the same type are combined to one connection and their size represents the quantity of combined relations.

In order to better analyze the relations the user can control the visualization:
- only relations of a specific type are shown
- only relations which are connected to a specific entity are shown
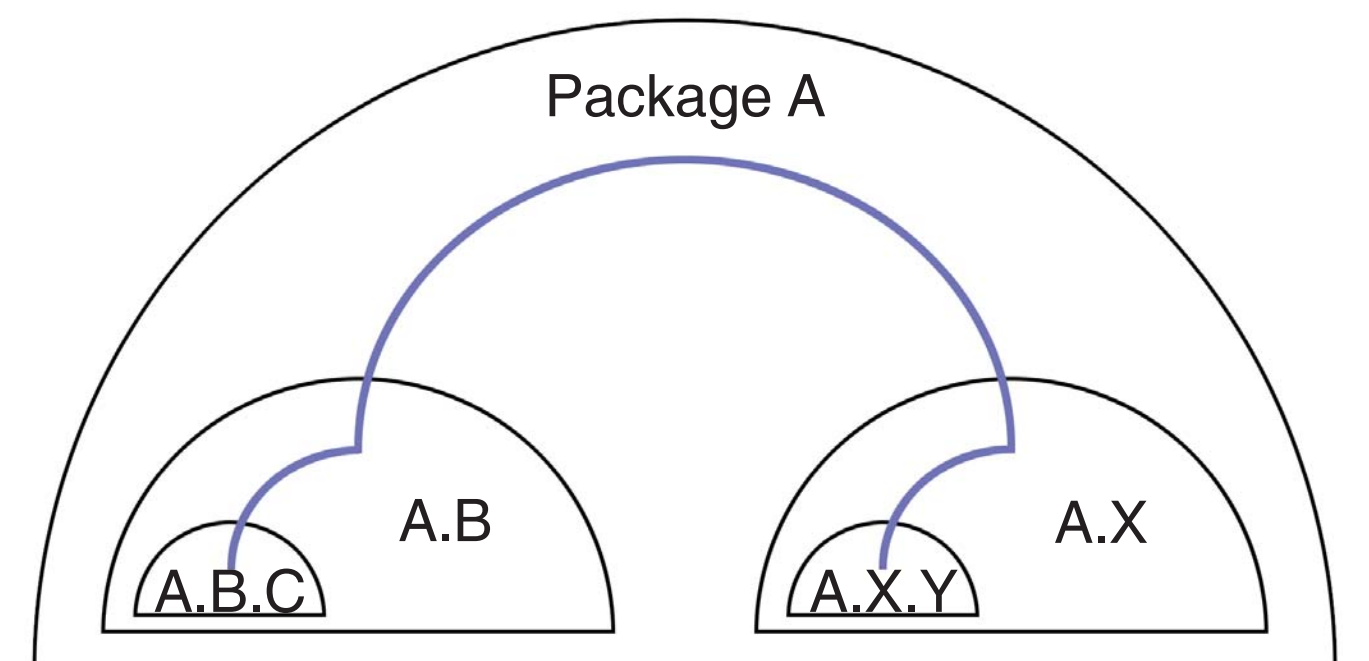
Routing: A.B.C ➔ A.B ➔ A ➔ A.X ➔ A.X.Y
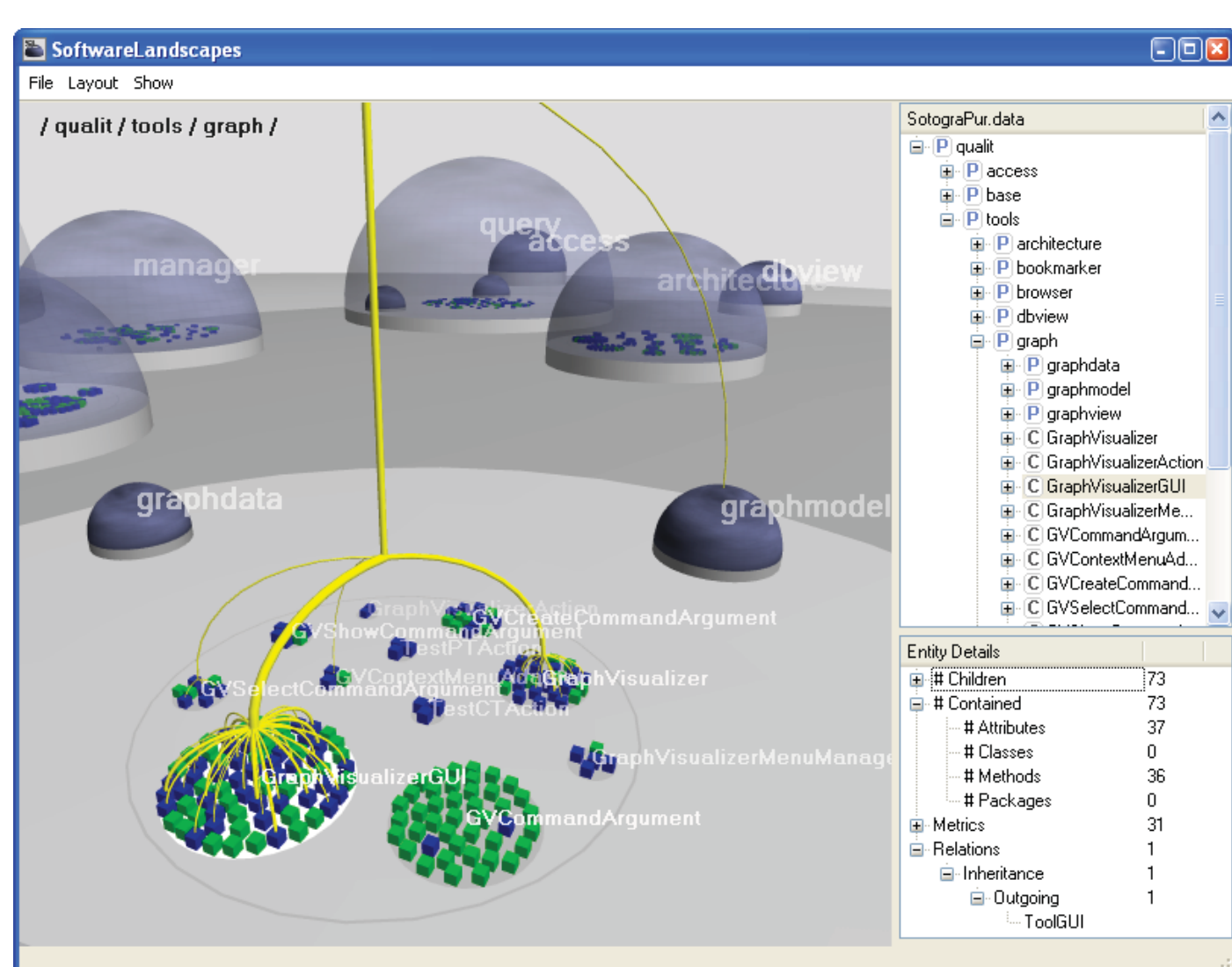
Figure 5: Relation routing in Hierarchical Nets

4 Utilizing Transparency for Level of Detail

The use of transparencies in the representation of packages (hemispheres) reduces the amount of presented information.

The degree of transparency is not fixed, but adapts dynamically to the position of the viewer by an infinitely variable cross fading.

Fading out distant levels allows to clearly represent also scenes with a very deep hierarchy.

The rendering performance of the visualization is also improved, because the inside of completely opaque hemispheres does not have to be drawn, and normally more than 90 percent of all hemispheres are completely opaque.

Figure 6: The application window

Figures 7-12: Zoom into the ‚Eclipse‘ open source platform, containing 112'613 entities and 339'161 relations, specified in 1.2 million lines of code