# Towards Seamless Semantic Zooming Techniques for UML Diagrams

Mathias Frisch*, Raimund Dachselt†
Computational Visualistics/Software Engineering Group
Otto-von-Guericke-University Magdeburg, Germany

Tobias Brückmann‡
Applied Telematics/e-Business Group
University of Leipzig, Germany

## Abstract

Models become increasingly important for software development processes. Though there is a multitude of software modeling tools available, the handling of complex UML diagrams is still difficult. In particular, the visualization of a global overview and of logical interrelationships between certain elements arising from refinements of diagrams can be improved. We address these problems and propose the usage of *semantic zooming* with different levels of detail and describe intuitive interaction techniques to ease the navigation between different diagrams in software models.

**CR Categories:** H.5.2 [Information interfaces and presentation]: User Interfaces—Graphical User Interfaces; D.2.2 [Software Engineering]: Design Tools and Techniques—User Interfaces

**Keywords:** UML, models, diagrams, interaction techniques, visualization, zoomable user interface, semantic zooming

## 1 Introduction

Software models can be huge, complex and hard to conceive as a whole. The unified modeling language (UML) is the most commonly used notation to design software systems. Models contain several UML diagrams of various types providing different views on the same system. Concerning diagrams, there are the following main requirements: Users want to get both, a reasonable overview of the system and a detailed view of certain diagram elements. Beyond that they want to understand logical relationships not expressed by UML between diagrams of different type. Widespread UML modeling tools use simple geometric zooming to navigate from overview to detail and vice versa. With this technique elements become very small and labels are unreadable. When zoomed in, navigation is often cumbersome as there is no context information available. Beyond that, there is usually one page for each diagram with just a single page visible at once. Users can navigate between diagrams by means of drop down menus, tabs or hyperlinks which change pages with a hard cut. These diagrams are visually isolated, although they are related on a logical level. We address these problems by using zoomable user interfaces (ZUIs). The contribution of this paper is twofold. On the one hand we suggest *semantic zooming* with different levels of detail (LoD) to interact with UML diagrams (see Fig. 1). On the other hand we present intuitive interaction techniques for ZUIs to improve the navigation in huge software models.

*e-mail: mfrisch@isg.cs.uni-magdeburg.de
†e-mail: dachselt@isg.cs.uni-magdeburg.de
‡brueckmann@ebus.informatik.uni-leipzig.de

## 2 Related Work

There are several approaches using ZUIs for interaction diagrams. In [Turetken et al. 2004] a fisheye view technique on data flow diagrams and ER-diagrams is presented. It enables the user to see certain processes on a finer level of granularity and the content of clustered entities respectively. Beyond that, approaches such as [Jacobs and Musial 2003] or [Koeth and Minas 2002] describe *semantic zooming* and focus and context techniques to navigate in package and class diagrams. A similar approach is implemented in ShriMP [Storey et al. 2002] which is an application for visualizing nested graphs and can also be used to explore ontologies and data flow diagrams. All the mentioned approaches try to solve the problem of limited overview and focus mainly on static diagrams (except for data flow diagrams). The interaction with behavior diagrams and the visualization of diagrams of different type nested into each other are not considered. An approach for the latter is presented in [Reinhard et al. 2008]. Here a special notation - not UML - is used to nest diagrams into each other. Besides that, this approach does not use *semantic zooming* with different LoDs, since elements are always shown in the same representation, and also lacks intuitive and efficient interaction techniques.

## 3 Semantic Zooming of UML Diagrams

*Semantic zooming* offers the possibility that elements of the respective diagram are displayed at different LoDs while zooming in and out. The amount and kind of information visible on a LoD varies depending on the available space and the semantic context.

### 3.1 Visualizing Overview and Details

The advantage of *semantic zooming* with different LoDs in comparison to a simple geometric zoom consists in providing an overview with important parts still visible. Therefore, the user is not confronted with small unreadable elements and can be supported in searching of particular parts of a diagram. For example, on an overview level of class diagrams classes are drawn as simple rectangles and just their labels and relationships are shown. No internal attributes, methods or cardinalities are visible. By zooming in, classes are expanded and methods and attributes appear. Fig. 3 illustrates that for the class 'Customer'. For more detailed examples please see [@CVSE]. Fig. 2 illustrates this technique applied to swim lanes of an activity diagram. On the coarsest level just the titles of the lanes are visible. By clicking on a certain swim lane

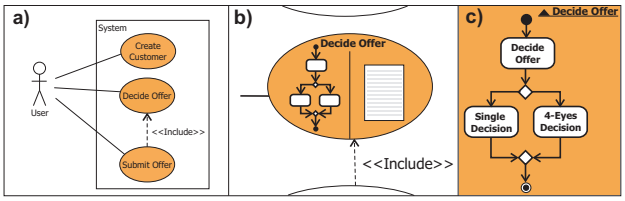


**Figure 1:** *LoDs of a use case diagram, 'Decide Offer' is further refined by a detailed text description and an activity diagram*

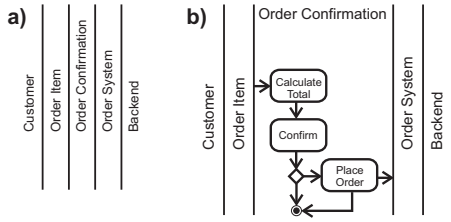


**Figure 2:** *Focus & context techniques for an activity diagram*

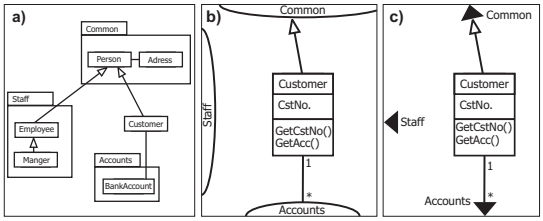


**Figure 3:** *Halos (b) and arrows (c) applied to a class diagram*

the lines move apart and the associated activities and transitions are displayed. A similar kind of visualization is also conceivable for the life lines of sequence diagrams. In this example *semantic zooming* is not performed on the whole diagram at once but just on a certain selected element. This is equivalent to a focus and context technique, as elements not in focus do not change their LoD. Of course it is also conceivable to apply a multi-focus technique with more than one element selected and zoomed in.

To ease and speed up the selection of many elements, we suggest circling the respective parts of a diagram (with mouse or pen)as an alternative to standard selection methods. Every circled area represents a focus and its LoD is increased. In conjunction with multi-focus visualization we also suggest a functionality which we call *zoom-to-filter*. This means elements are selected according to certain criteria, such as “last edited” or “connected with” and are zoomed in automatically. These techniques can especially speed up selection processes in huge diagrams, as elements do not have to be selected and de-selected one by one.

When zoomed in on a huge diagram, navigating from one area to another by simple panning can be cumbersome. To reduce this effort and to provide a quick navigation to off-screen elements, we propose to use Halos [Baudisch and Rosenholtz 2003] or similar techniques. Fig. 3 depicts two examples applied to a class diagram. When zooming in on a particular class, Halos (b) or arrows (c) appear at the border of the screen. They can be displayed for the nearest neighboring elements, for those elements which are connected with the currently focused one or for clusters of certain elements which are generated according to semantic criteria. Clicking on one of these navigation aids starts an automatic panning to the associated elements.

### 3.2 Visualizing Refinements

The refinement of model elements by other artifacts plays an important role, in particularly in model driven development processes. Generally, diagrams are nested into each other. ZUIs can be used to visualize these nestings in a more comprehensible way and allow a quick navigation by smooth animated transitions. Fig. 1 shows different LoDs for a semantic zoom of use case diagrams. By zooming in on a particular use case, a preview of nested elements is given (b). Users can choose between a nested activity diagram and a textual description, both describing the use case in more detail. The result of zooming on a nested activity diagram is shown in Fig. 1 c). Other examples for the nesting of UML diagrams are the following: deployment diagrams can be refined by component diagrams; component diagrams for their part can be specified in detail by package or class diagrams; and class diagrams can be refined by state charts or source code. All these hierarchies of nestings could be navigated by means of *semantic zooming*, whereby zooming in can be accomplished with the mouse wheel or by clicking on the respective element, and zooming out can be done by clicking on the background. One problem for the user is to realize on which level of the hierarchy she is, or if the diagram currently visible is nested into another one. To prevent this, we suggest links by the side of the diagram which lead to the superior element (see Fig. 1 c)) or buttons at the border of the screen which allow a quick navigation to higher or lower levels. Another beneficial feature can be color coding: e.g., use cases are orange. So the background of the nested activity diagram is shaded orange too, symbolizing that this diagram is nested in a use case.

## 4 Conclusion & Future Work

We presented animated *semantic zooming* techniques including different LoDs for UML diagrams. With it, overview and the comprehension of interrelated diagrams can be improved. Furthermore, we suggested UI techniques which can support the navigation in huge diagrams in conjunction with ZUIs, such as multi-focus techniques, Halos and diagram refinements. By now we built low-level prototypes to demonstrate the proposed concepts. Animations and pictures further illustrating the concept can be found in [@CVSE]. Currently we are working on implementations of these visualization and interaction techniques as high-level prototypes. Next steps will include user studies to compare state-of-the-art tools to our approach, concerning accuracy, speed and comprehension.

**Acknowledgements** This work was partly funded by the ”Stifterverband für die Deutsche Wissenschaft” from funds of the Claussen-Simon-Endowment.

## References

BAUDISCH, P., AND ROSENHOLTZ, R. 2003. Halo: a technique for visualizing off-screen objects. In *Proc. of CHI ’03*, 481–488.

@CVSE: University of Magdeburg, Germany, supplemental Material for this paper, http://isgwww.cs.uni-magdeburg.de/cvse/Forschung/SZofUML/.

JACOBS, T., AND MUSIAL, B. 2003. Interactive visual debugging with uml. In *Proc. of SoftVis ’03*, 115–122.

KOETH, O., AND MINAS, M. 2002. Structure, abstraction, and direct manipulation in diagram editors. In *Proc. of DIAGRAMS ’02*, 290–304.

REINHARD, T., MEIER, S., STOIBER, R., CRAMER, C., AND GLINZ, M. 2008. Tool support for the navigation in graphical models. In *Proc. of ICSE ’08*, 823–826.

STOREY, M.-A., BEST, C., MICHAUD, J., RAYSIDE, D., LITOIU, M., AND MUSEN, M. 2002. Shrimp views: an interactive environment for information visualization and navigation. In *CHI ’02: extended abstracts on Human factors in computing systems*, 520–521.

TURETKEN, O., SCHUFF, D., SHARDA, R., AND OW, T. T. 2004. Supporting systems analysis and design through fisheye views. *Commun. ACM 47*, 9, 72–77.