UNIVERSITY OF GOTHENBURG

3D Visualization for Model Comprehension

A Case Study Conducted at Ericsson AB

ANNE-KATRIN KROLOVITSCH
LINDA NILSSON

Bachelor of Applied Information Technology Thesis

Report No. 2009-047
ISSN: 1651-4769

Academic Supervision: Lars Pareto (Gothenburg University)
Industrial Supervisors: Staffan Ehnebom, Magnus Standar, Krister Bergh (Ericsson)

University of Gothenburg
Department of Applied Information Technology
Göteborg, Sweden, May 2009

3D Visualization for Model Comprehension — A Case Study Conducted at Ericsson AB

Anne-Katrin Krolovitsch
IT University of Gothenburg
Software Engineering and Management
Gothenburg, Sweden
annekatr@ituniv.se

Linda Nilsson
IT University of Gothenburg
Software Engineering and Management
Gothenburg, Sweden
nilind@ituniv.se

Abstract

With growing size and complexity of models, model comprehensibility has become a problem in model driven development. Common experiences are information overload in diagrams and difficulties in understanding model hierarchies, both leading to users spending too much time on comprehending models. The paper presents a case study at a department at Ericsson AB, experiencing this problem. The case study investigates the problem through iterative interviewing and incremental prototyping of mixed fidelity prototypes of a 3D model visualizer. Prototypes include structure diagrams, sequence diagrams and state machines — all recurrent in model driven development tools. Findings show that comprehensibility is improved by visualizing model hierarchies in 3D, by providing filtering for diagrams, by expand/collapse operations on the model's structure, and by utilizing the 3D space for visualizing components as 3D boxes and pipes. Feedback received from informants show that moving from a 2D perspective to a 3D perspective has substantial impact on model comprehension.

Categories and Subject Descriptors D.2.2 [*Software Engineering*]: Design Tools and Techniques: Computer-aided software engineering (CASE); D.2.2 [*Software Engineering*]: Design Tools and Techniques: User Interfaces

Keywords MDD, RSARTE, 3D Visualization, Model Comprehension

1 Introduction

Model driven development is an approach to software development that aims at giving the software developers a better understanding of the software built, by raising the level of abstraction in programming from textual code to visual models. This reduces the gap between design and implementation, which makes the software built easier to comprehend.

However, many model driven development tools [9, 10, 4, 8] do not fully deliver on their purpose. Although today's tools indeed raise the level of abstraction, they put high demands on user cognition, preventing better understanding that model driven development is supposed to bring. Comprehending complex models, using today's tools, involves manual non-trivial traversing of the model's hierarchy (i.e. its components and subcomponents). This puts high demands on the user to develop a strategy to understand and memorize a model's structure. Furthermore, in today's model driven development tools, model comprehension is made difficult through information overload in overly large diagrams. Many diagrams contain overwhelmingly much information which makes it challenging for the user to understand the diagrams structure.

These problems of model comprehension have been experienced by a department of Ericsson AB, located at Lindholmen in Gothenburg. Ericsson is a leading provider of telecommunication and data communication systems for the international market. The department has made use of the model driven development approach for years and has, with several generations of modeling tools, been able to successfully design software for large and complex embedded systems. The problems experienced concerning the tools are the following: difficulties with understanding a new model and learning its structure; difficulties with navigating through the model; difficulties with understanding the diagrams in the models.

The purpose of this paper is to improve the comprehensibility of large and complex models, as found in projects utilizing model driven development at a large scale. The scope of improvement is structure diagrams, sequence diagrams and state machines (also known as state diagrams), as these have a predominant role in model driven development.

The paper's approach is to take the visualization of the hierarchical structure in these types of diagrams from a 2D to a 3D perspective, bypassing the limited expressiveness of the 2D space. This approach is supported by Louis Feng et al., where they argue that "using one view [2D] of the data limits the number of attributes and the available exploration space" [14].

To explore the 3D perspective, iterative prototyping was used on the basis of existing 2D models. These 2D models were created in a previous research project in the modeling tool Rational Software Architect RealTime Edition [8], which we from here on will refer to as RSARTE. We will present design solutions for a 3D visualizer, in the form of non-functional prototypes created in Google SketchUP [11].

Our results show that going from 2D to 3D will enhance the comprehensibility of complex models and open up for the opportunity to filter visible information. The idea, to go from 2D to 3D, to improve model comprehension for state machines has previously been explored by Paul McIntosh et al. [15]. State machine diagrams, created in Rational Rose RealTime [4], were visualized in 3D and users were asked to judge the usability of 3D visualized diagrams. The result of their usability test showed that the users expressed that the tool had high usability. Furthermore, our results show that the problem of information overload in diagrams can be improved by offering dynamic filtering options, giving the opportunity to reduce the amount of visible data by hiding selected parts of the diagrams.

The precise research question addressed in this paper is as follows:

Q1 How can sequence diagrams, structure diagrams and state machines, used with RSARTE, be visualized in 3D, focusing on the visualization of model hierarchy and information filtering?

The paper is organized as follows: in chapter 2 we present the research design for our study, chapter 3 explains the outcome and results of this research, chapter 4 includes a discussion about the applied techniques, future work as well as related work, and in chapter 5 we present our conclusions.

2 Research Design

To address the research question, we have conducted a single case study. In a case study "the researcher explores in depth a program, an event, an activity, a process or one or more individuals" [5]. It is also said that "although they [case studies] cannot achieve the scientific rigor of formal experiments, case studies can provide sufficient information to help you judge if specific technologies will benefit your own organization or project" [12]. The case under investigation is a specific problem to the department at Ericsson, where this case study was conducted using a qualitative research approach.

As a method for investigating the research question, we have chosen to perform four semi-structured interviews, each conducted with one employee at Ericsson. The informants are software developers, with different levels of work experience in model driven development, ranging from one to six years. The model driven development tool of their daily use is Rational Rose RealTime. Our investigation of the research question is based on the qualitative analysis [5] of the data collected from these interviews.

2.1 Process

In the tradition of human-computer interaction research, we recognize three fundamental activities in design: understand the requirements, produce a design that satisfies those requirements, and evaluate the design [20]. Our research process is based on three iterations over these activities (see Figure 1). The first iteration consists of *Initial Phase, Prototypes, Interviews, Transcription,* and *Analysis*. The second iteration includes *Requirement Specification, Prototype Updates,* and *Validation Interview*. The third iteration consists of *Requirement Specification Update* and *Final Prototypes*.

A key advantage of this process is that it engages end users in the design, gives quick feedback on design solutions from end users, and allows for development of realistic and valid prototypes.

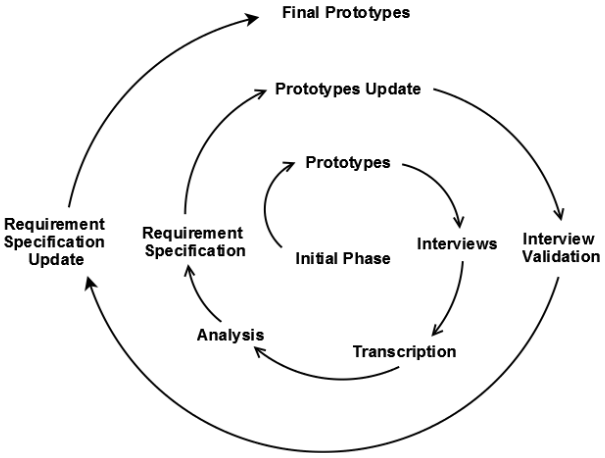

Figure 1. Research process

Initial Phase

During the initial phase of the research process, we gathered information of research previously conducted in the area of 3D visualization, with focus on UML diagrams. The purpose of this was to see if similar work existed, in which case we could build upon that research. As part of this phase we investigated RSARTE and had brainstorming sessions and discussions during which we spawned ideas for our first prototypes.

Prototypes

The 2D models which the prototypes are based on, are representative for the models used at the department at Ericsson AB. The prototypes were created in Google SketchUp, which has good support for quick creation of realistic 3D prototypes. These were used as a basis for the interviews of the first cycle.

Interviews

The interviews were performed to get insights into the users' views on our prototypes and the 3D visualization techniques applied. The interviews were conducted and recorded to collect suggestions for improvement of the prototypes and to gather requirements for future implementation of 3D visualizations.

Transcription

To be able to conduct a qualitative analysis, the interviews were transcribed. This gives more accurate and complete data than taking notes during the interviews. For the transcription we used the software Express Scribe [17].

Analysis

For the analysis of these interviews we applied the qualitative analysis techniques, coding and categorization [5]. These are supported by the Atlas.ti analysis software [2] which we used for this activity. The first step of the analysis was to organize and interpret the interview transcriptions. After this we used a coding process to label the findings in the transcriptions. This was followed by grouping the codes into different categories. The purpose of these qualitative analysis techniques was to define the requirement areas in the requirement specification.

Requirement Specification

The result of the analysis was presented in a requirement specification document. More specifically, it was presented in a vision document, as of the Rational Unified Process [7]. This was because a requirements specification is an intuitive and lucid way for presenting the findings to engineers.

Prototypes Update

After the requirement specification activity, we went back to the prototypes; according to our findings in the requirements specification, we updated the prototypes in SketchUp.

Interview Validation

To validate the implementation of the changes in the prototypes as well as the collected requirements, we asked all informants for a validation meeting, during which we discussed the requirement specification document as well as the prototypes. For both of these, the informants had minor change requests.

Requirement Specification Update

As part of the last iteration we finalized the requirement specification document for the 3D visualizer.

Final Prototypes

In the last activity of the process we updated the prototypes according to the updates of the requirement specification.

3 Results

3.1 Problem areas in 2D

As part of the initial phase, before the first prototypes were built, we identified different problem areas in models visualized in 2D concerning model comprehension. These areas were found by investigating the modeling tool RSARTE. The following problem areas were identified:

- Visualization
- Navigation

- Filtering

One major problem area in model comprehension is the 2D visualization in today's tools, which only offers the possibility to view one diagram per window and have the referenced diagrams open in tabs. This problem was confirmed by an informant stating that ”in today's situation it is difficult to understand complex models this way”. For example, large diagrams can have more than 300 references in the hierarchy, where one subcomponent can be referenced by several other components, which is not visible in the 2D view.

Another major problem area is navigation, due to that only one diagram per window can be seen. This puts major constraints on the navigation through the hierarchy of a model. The user needs to traverse in the modeling tool by clicking a reference (e.g. a state in a state machine) to open the referenced diagram in a new window, which also limits the comprehension of complex models. Informants confirmed this problem, saying that ”one easily can loose overview of the model, not remembering the structure and design of the model”. This supports that the lack of navigation techniques in 2D puts high demands on the user to memorize the model.

The third major problem area is information overload due to a lack of filtering functionality. Complex diagrams cannot be dynamically abstracted to only show the parts of the diagrams which are currently important to be visible.

3.2 Requirements Specification for 3D visualizations

As described earlier, the found requirements were collected in a requirement specification for the 3D visualizer. The collection includes requirements gathered from the Initial Phase, Interviews and Interview Validation activities. The requirements in Table 1 show all requirements applicable for structure diagrams, state diagrams and sequence diagrams. As can be seen in the table, they are divided into five different groups of requirements, three of which reflect the problem areas identified in section 3.1. The other two groups, ”2D/3D Mode” and ”Search” were discovered during the interviews.

The first area, ”Visualization”, lists requirements which will improve the comprehension of a model by visualizing its hierarchy (R2). Also, partial visualization of a model shall be possible (R1), where the user can select what parts of a model should be visualized in 3D.

The second area, ”Filtering”, presents the requirements for improving information overload in the diagrams and in the model. Therefore, dynamic filtering in these shall be made possible (R9). It shall also be possible to save filter settings as bookmarks (R11) so that the user easily can switch between filtrations that he or she has used in the past.

The third area is ”Navigation” and holds requirements regarding how model comprehension can be improved by enhancing navigation techniques. These shall be similar to the navigation in a Computer Aided Design (CAD) tool [3], where the user can zoom, pan, and rotate the model (R22-R24).

The fourth area is ”2D/3D mode” which holds the requirement for the possibility to switch between the 2D view used today and the 3D view we propose (R27), being able to switch between the modes will allow the user to use 3D to understand the model's structure and use the 2D mode when working with details in a diagram. This area has a requirement stating that it should be possible to edit the models in the 3D mode (R29).

The fifth area is ”Search”, which deals with how the search result can be presented for the 3D mode. If there is only one hit, that result should be put into focus in the 3D visualization (R33). However, if there are many hits, they should be presented in a hierarchical list (R32), where the results are separated into two groups: one group listing the hits found in the 3D view and the second group listing all hits found in the remaining part of the model.

During the interviews the informants were asked which other parts of their daily work could be made easier and more intuitive with 3D visualization. In addition to the requirements for the stated types of UML diagrams, the informants expressed ideas for 3D visualizing use case diagrams, data classes, packet diagrams and the version control system.

Requirements for Structure, State and Sequence Diagrams

Visualization

R1 Partial visualization of a complex model shall be possible

R2 Referenced and Referencing diagrams shall be visible

R3 Collapsed parts shall be made visual with a symbol

R4 Every diagram shall have a label with its name

R5 Every reference shall have a label stating referenced and referencing diagrams

R6 The diagram itself shall be visualized in 3D, using the space more efficiently

R7 Structure diagrams and state diagrams shall be visualized in the same view

Filtering

R8 Default filter settings shall be available

R9 The user shall be able to dynamically filter on any terms needed

R10 It shall be possible to save own filter settings

R11 Several filter settings shall be saved as bookmarks

R12 It shall be possible to collapse hierarchies

R13 Expanding the collapsed hierarchies shall be done by double clicking on the symbol (see R3)

R14 In a diagram it shall be possible to hide parts(objects, lifelines, references, labels, etc.)

R15 In sequence diagrams it shall be possible to collapse and expand fragments (alternatives, options, loops, etc.)

R16 Dynamically hide and view objects in sequence diagrams shall be possible

Navigation

R17 Classical file tree, visualizing the model hierarchy, shall be available

R18 Rocket button shall give the user the bird's-eye view of the model

R19 Getting a diagram in focus shall be done by double clicking on it

R20 Via a keyboard combination it shall be possible to switch focus between referenced and referencing diagrams

R21 3D Navigation controls shall be accessible via keys on the keyboard

R22 One shall be able to rotate the model

R23 One shall be able to zoom in on the model

R24 It shall be possible to pan to see the model without changing angle nor position

R25 Select and horizontally move a diagram in relation to the others shall be possible

R26 Navigation history shall allow users to switch between diagrams

2D/3D Mode

R27 Switching between 2D and 3D mode shall be possible

R28 To get the top view of a diagram in 2D the user shall be able to zoom in on 2D from 3D

R29 One shall be able to edit models in 3D; the changes shall be visible in 2D as well

Search

R30 A search function shall be available

R31 When a search is conducted all occurrences related to the model element shall be highlighted

R32 Search results shall be presented in a hierarchical list

R33 When search result has only one hit, that one hit shall be put into focus

Table 1. Requirements table

3.3 Explorations in 3D visualization

3.3.1 Initial 3D Prototypes

The outcome of the initial prototype activity (see section 2.1) is shown in figure 2–4.

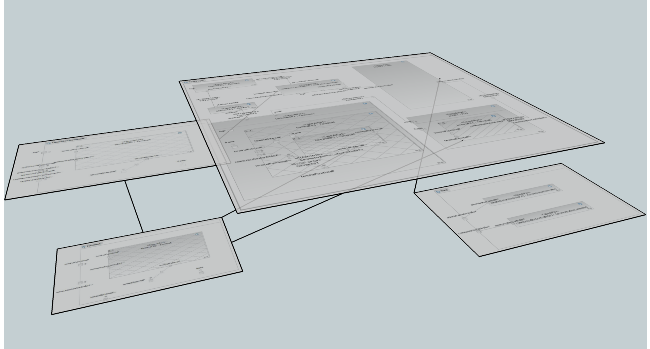

Figure 2. Structure Diagrams

Figure 2 shows a 3D visualization of a set of structure diagrams. This image shows that the user can see the referenced diagrams immediately and does not have to view one diagram at a time to understand the model's structure.

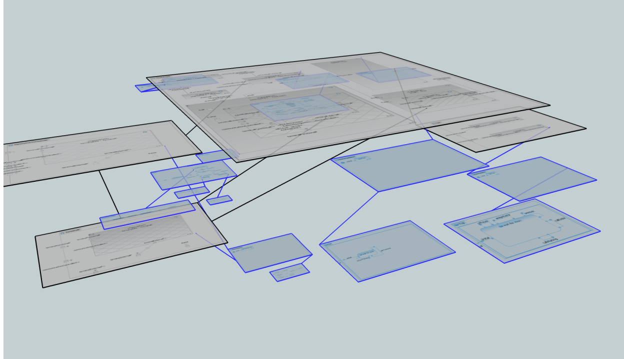

Figure 3. Structure and State Diagrams

In the prototype displayed in Figure 3, structure diagrams and their state machines are visualized in 3D. Both types of diagrams are visualized in the same view, because during the initial phase, the need for this solution became obvious (see R7 in Table 1). Since the navigation in the 3D visualizer will work like a CAD tool (see R22-R25, Table 1), it will be possible through the zooming functionality to see the diagram closely. The rotating functionality will allow the user to spin the 3D model and look at it from different angles. The panning functionality will allow the user to see other parts of the 3D model, or to horizontally move selected diagrams.

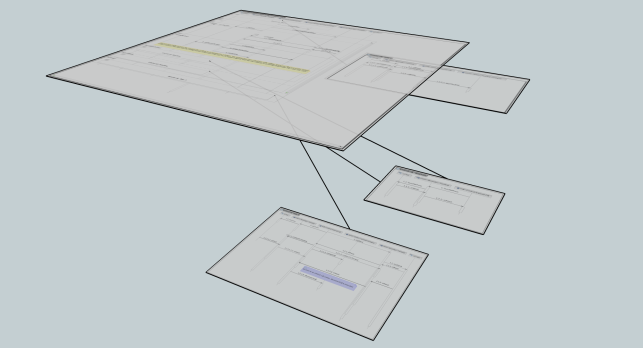

Figure 4. Sequence Diagrams

The diagrams in Figure 4 are sequence diagrams. This 3D view is an example of how references in sequence diagrams can be visualized. Also here, through the functionalities of a CAD tool, it will be possible to see the diagrams closely, rotate them, and pan them. In sequence diagrams there will be the possibility to collapse and expand fragments (e.g. alternatives, loops, options) according to R15. This will give the user the freedom to choose which parts of the sequence diagram should be visible.

3.3.2 Final 3D Prototypes

This section presents the updates that were applied to the final prototypes as well as the reasons for these updates.

The updates of the prototypes were based on requirements R3-R6 in Table 1. The idea for requirement R6 (The diagram itself shall be visualized in 3D, using the space more efficiently) emerged through discussions during the requirement specification phase in the second iteration. The idea was to not only visualize the diagrams as different levels in a 3D hierarchy but also to draw the components in the diagrams in 3D to make better use of the space. This allows the user to see nested components from a side view. The changes made to the prototype of the structure diagram are shown in Figure 5, where the components are modeled as boxes and the references between the components are modeled as pipes. These components are concurrent classes, commonly known as capsules [16], which communicate using a specified protocol. The same change was made for states in state machines as depicted in Figure 6. These visualizations of the diagrams were well received at the validation meeting with the informants, who thought that seeing the nested components from the side gave the possibility to quickly grasp the complexity of the model. For the sequence diagrams, two different solutions

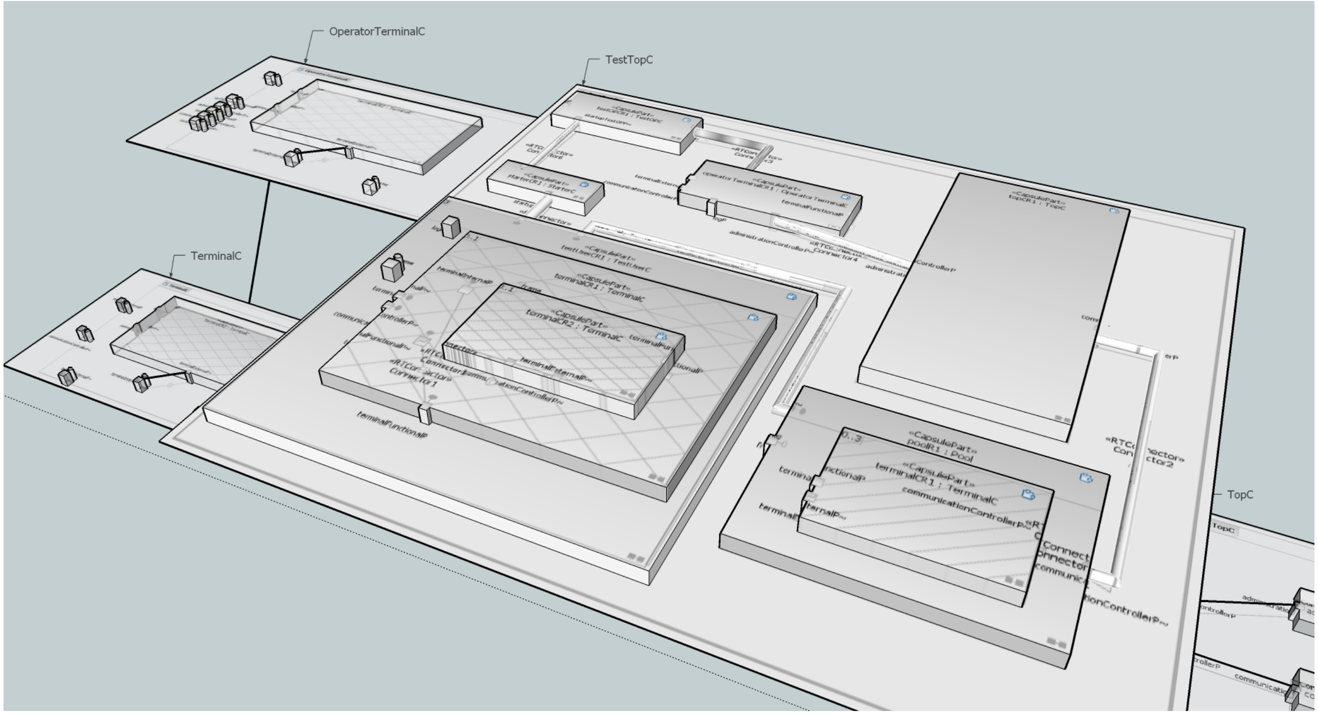

Figure 5. Structure Diagrams modeled with 3D boxes

were presented at the validation meeting (see Figure 8 and Figure 9). However, as we suggest under future work, the area of modeling sequence diagrams as 3D boxes needs further investigation (see section 4.2).

In the updated diagram for R3 (Collapsed parts shall be made visual with a symbol), the symbol is displayed as a cube (see Figure 7). The symbols in this figure represent state diagrams which are filtered out (see R12, Table 1), this is done to emphasize that there is a reference but giving the user the option to not display its content. By double clicking the cube, the hierarchy will be expanded again (see R13, Table 1).

Requirement R4 (Every diagram shall have a label with its name) was added due to an informant saying that ”you could add a text so that one clearly can see what it is. Sometimes you remember the name of a state and then you would see it faster, instead of remembering what it looked like”. The updated prototype is displayed in Figure 5. According to R9 (The user shall be able to dynamically filter on any terms needed) one will be able to hide the labels when they are not needed.

Finally, a label was added to the reference line according to requirement R5 (Every reference shall have a label stating referenced and referencing diagrams). In a discussion during the validation meeting, there was a request to display the label with the name of the referenced diagram as well as all referencing diagram names higher in the hierarchy, as shown in Figure 6. This was to help the user navigate in the model; one of the informants said that ”if the user knows the names of diagrams but not the structure, it would be a faster way to orient oneself in the model”. However, if the hierarchies are deep the names will be long. Therefore, it shall be possible to filter out these labels when they are not needed (see R9 and R14).

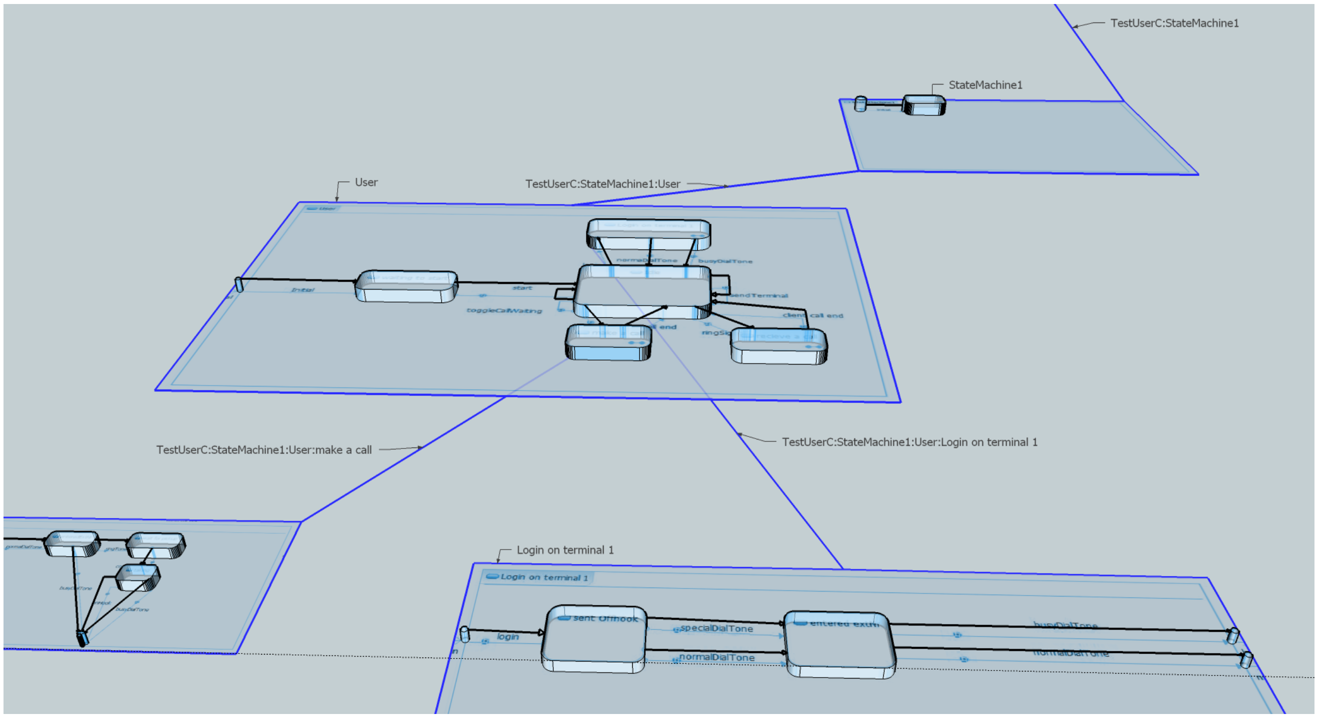

Figure 6. State Diagrams modeled with 3D boxes

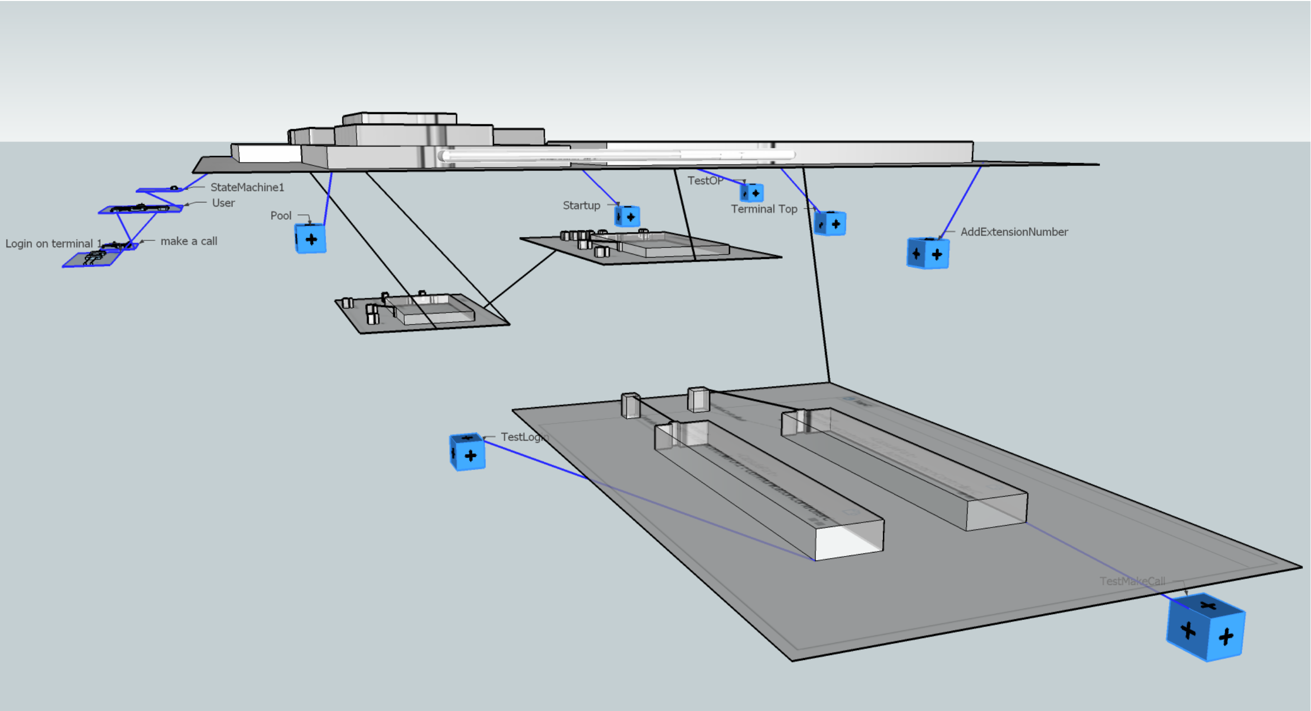

Figure 7. Symbol for hidden state diagrams

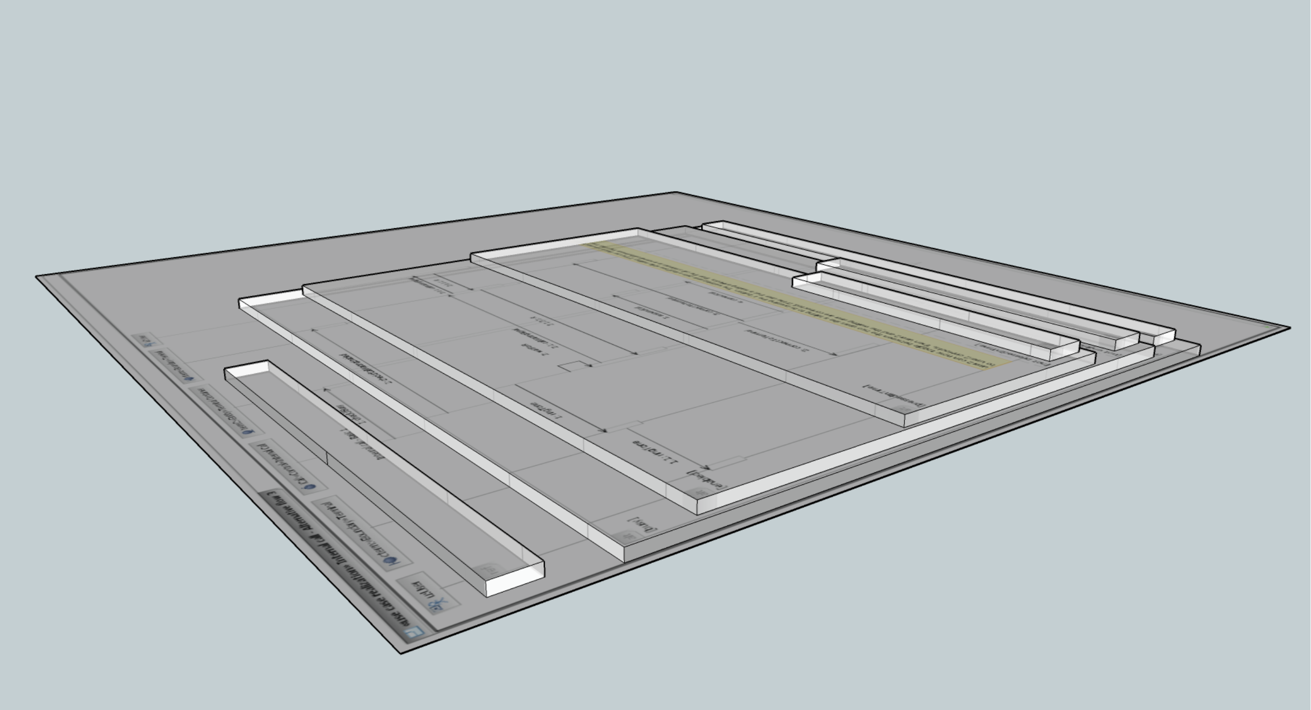

Figure 8. Sequence diagram alternative 1

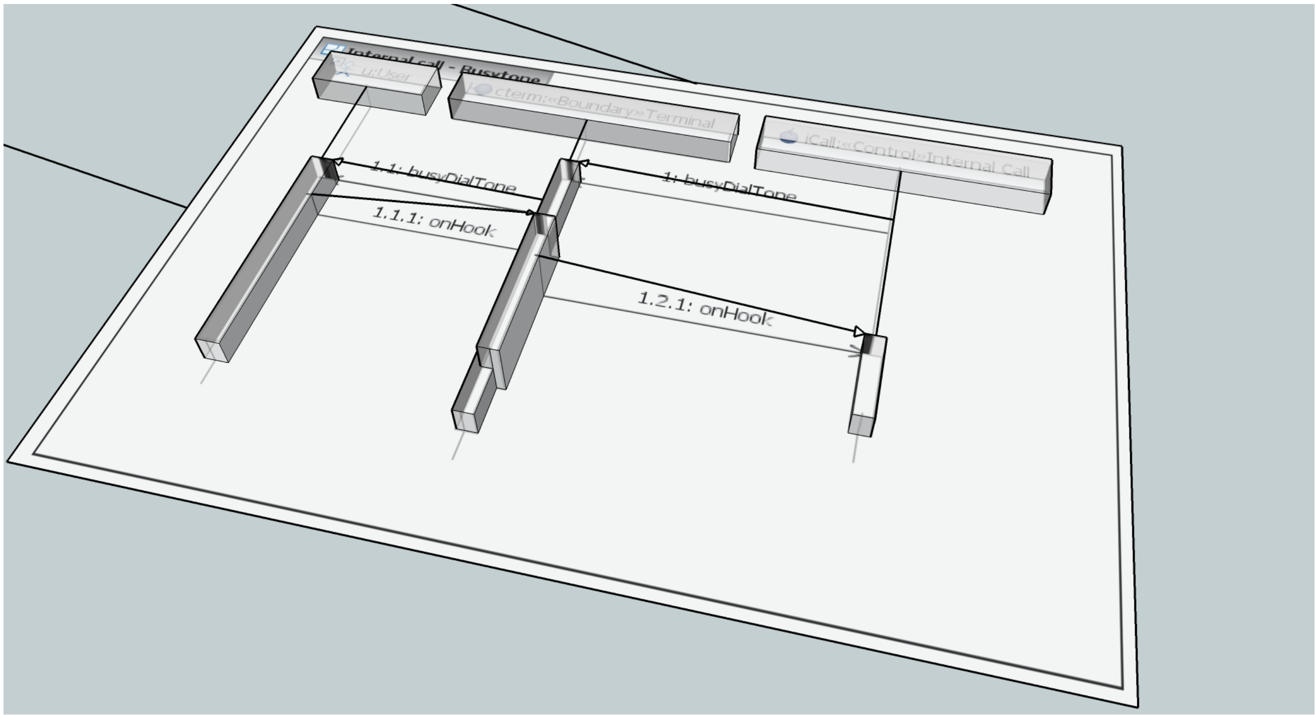

Figure 9. Sequence diagram alternative 2

4 Discussion

4.1 Prototypes

We strived to design parts of the 3D prototypes, which will enhance model comprehension, as realistic as possible. This had the effect that the prototypes hold qualities of both high and low fidelity prototypes. They are low fidelity because they are made of static diagrams and have no functionality. They are simple, cheap and quick to produce and modify; this means that they support the exploration of the design [20], which allowed us to easily change them and include new requirements. They are high fidelity prototypes, since they look like a possible final solution, which provides a balance to provisional paper prototypes [20].

The realistic prototypes helped to show the informants what structure diagrams, state machines and sequence diagrams will look like in 3D and how navigation in these representations will work. The informants showed strong enthusiasm about the idea to visualize the models in 3D: one informant said that 3D will benefit ”everyone who uses model based development and would like to see the structure of things”.

Using realistic prototypes also opened up for asking how the informants experienced the 3D environment: their response was that they found it intuitive and that it eased the understanding of the model’s hierarchy compared to a 2D view.

New ideas were conveyed for 3D visualization of other diagrams in RSARTE (e.g. Use Case diagrams, Data Classes and Package Diagrams). This shows that the concept of 3D visualization which we presented can be transfered to different types of diagrams.

The fact that our prototypes created such enthusiasm about 3D visualization was an interesting observation. Obviously, this type of visualization can be used for other tools than RSARTE as they handle the same type of diagrams as the ones we have investigated. Finally, this specific case study can also be generalized (through analytic generalization, see [19]) in that not only the department at Ericsson AB, which expressed the need for improvement in model comprehension, can benefit from such a 3D visualizer, but also other users of model driven development tools in the large.

4.2 Future work

One area of future work is the implementation of 3D visualizers. One way to implement such a visualizer would be in the form of a plug-in for Eclipse [1], which many model driven development tools, including RSARTE, are based on.

After the implementation of R6 (The diagram itself shall be visualized in 3D, using the space more efficiently), it became clear that visualizing the complexity of sequence diagrams needs further investigation. During the discussions with end users, it became apparent that besides the solutions we presented in our prototypes (see Figure 8 and Figure 9), the complexity of these type of diagrams can also be visualized in other ways. One way would be a visualization of the amount of nested references, by developing an algorithm which calculates the height of each box for each reference according to its complexity in terms of the number of objects. Another way would be to visualize complexity through the use of different colors.

Furthermore, if the suggestions for 3D visualization of use case diagrams, data classes, packet diagrams and the version control systems, as put forward by the informants, should be taken into consideration, then future research needs to be conducted in these areas as well. Special request for detailed research was conveyed for version control systems, which an informant pointed out to be one of the major problems with Rational Rose RealTime. This informant also expressed that it is important when designing the architecture for the 3D visualizer, that future extensions concerning the version control system need to be possible.

4.3 Related Work

The previously outlined problems within 2D visualization (see section 3.1) are not new in the area of model driven development. Paul McIntosh et al. [15] studied this problem in terms of 3D visualization of UML state machines in Rational Rose RealTime. In their research, they investigated whether there is a measurable benefit in 3D state machines as well as if it reduces cognitive load in state machine behavior compared to 2D modeling. From their measurements they concluded that the true benefit of 3D visualization ”is in the area of understanding hierarchical state machine diagrams”, since 3D allows the user to see the complete state machine in one view.

Also, the following works show that the interest in 3D visualization has increased over the last decade. In their work from 1999, Knight & Munro [13] researched comprehension within virtual environment visualization, discovering the possibility for code visualization. This idea was then further developed in 2003 by Louis Feng et al. [14].

Moreover, Richard Wettel and Michele Lanza [18] have developed an interactive 3D visualization tool, where systems, classes, and packages are visualized through a 3D city metaphor. This tool supports the analysis of large-scale object oriented software systems.

In 2001, Tim Dwyer [6] conducted a usability study examining ”how class and object diagrams [...] can be better understood through 3D visualization”. He claims that using 3D enabled the user to identify possible problem areas of strong coupling and low cohesion in software architectures. As a result of the study he concludes that ”an effective method of layout for 3D UML diagrams” has been presented which ”improves a user’s cognition of the architectural structure of complex system models”.

5 Conclusion

The result of our study shows how sequence diagrams, structure diagrams and state machines can be visualized in 3D. This can be done through visualizing diagrams and the relation between the diagrams (also referred to as hierarchy) in one view, as well as visualizing different diagram types and their relation in the same view. The possibility to view hierarchies is strengthened through navigation techniques commonly used in CAD tools (rotate, pan and zoom). This allows the user to see the relations between diagrams from different angles. Another way to show the hierarchies is by placing labels on the references between diagrams. These specify the referenced diagram as well as all referencing diagrams in the hierarchy above.

Filtration alternatives are made possible in the 3D prototypes, solving the problem of information overload in complex diagrams and models. We have shown how hierarchies can be collapsed and expanded according to user needs. Furthermore, the design of our solution also proposes that in the future 3D visualizer it will be possible to filter within the diagrams, as for example in sequence diagrams where fragments (options, alternative, loops, etc.) can be collapsed.

During this study we discovered that 3D visualization can be taken one step further through elevating individual components within each diagram, making them look like boxes. Using the available space has the benefit that also the complexity of diagrams can be visualized in 3D. Due to that nested components are visualized as stacks of boxes, complexity in the diagrams is visible from a side view.

The response received about the prototypes during interviews, shows that the way we visualize sequence diagrams, structure diagrams and state machines in 3D allows for a better comprehension of complex models in model driven development.

Acknowledgments. We would like to thank our academic supervisor Lars Pareto, as well as our industrial supervisors Staffan Ehnebom, Magnus Standar, and Krister Bergh for their support and valuable discussions through out this research. Also, we would like to thank Henric Stenhoff who helped to make this research possible, as well as the anonymous informants who volunteered their time for the interviews.

References

[1] Eclipse. http://www.eclipse.org/, 2009-05-29.

[2] Berlin ATLAS.ti GmbH. atlas.ti the knowledge workbench. http://www.atlasti.com/, 07.05.2009.

[3] Articles BPC and Glossary. Cad: Computer-aided design. http://www.bestpricecomputers.co.uk/glossary/computer-aided-design.htm, 2009-05-27.

[4] Rational Software Corporation. Rational rose realtime. http://www.ghs.com/partners/rational/rose-rt.pdf, 19.05.2009.

[5] John W. Creswell. *Research Desing Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications, Inc., second edition, 2003.

[6] Tim Dwyer. Three dimensional uml using force directed layout. In *APVis '01: Proceedings of the 2001 Asia-Pacific symposium on Information visualisation*, pages 77–85, Darlinghurst, Australia, Australia, 2001. Australian Computer Society, Inc.

[7] IBM. Rational unified process. http://www-01.ibm.com/software/awdtools/rup/?S_TACT=105AGY59&S_CMP=WIKI&ca=dtl-08rupsite, 07.05.2009.

[8] IBM. Rational software architect realtime edition, version 7.5.2. http://www-01.ibm.com/support/docview.wss?rs=3642&uid=swg27015122, 19.05.2009.

[9] IBM. Telelogic rhapsody. http://www.telelogic.com/products/rhapsody/index.cfm, 19.05.2009.

[10] IBM. Telelogic tau - model driven development of complex systems and services. http://www.telelogic.com/products/tau/, 19.05.2009.

[11] Google Inc. Google sketchup. http://sketchup.google.com/, 07.05.2009.

[12] Barbara Kitchenham, Lesley Pickard, and Shari Lawrence Pfleeger. Case studies for method and tool evaluation. *IEEE Softw.*, 12(4):52–62, 1995.

[13] Claire Knight and Malcolm Munro. Comprehension with[in] virtual environment visualisations. In *IWPC '99: Proceedings of the 7th International Workshop on Program Comprehension*, page 4, Washington, DC, USA, 1999. IEEE Computer Society.

[14] Andrian Marcus, Louis Feng, and Jonathan I. Maletic. 3d representations for software visualization. In *SoftVis '03: Proceedings of the 2003 ACM symposium on Software visualization*, pages 27–ff, New York, NY, USA, 2003. ACM.

[15] Paul Mcintosh, Margaret Hamilton, and Ron Schyndel. X3d-uml: 3d uml state machine diagrams. In *MoDELS '08: Proceedings of the 11th international conference on Model Driven Engineering Languages and Systems*, pages 264–279, Berlin, Heidelberg, 2008. Springer-Verlag.

[16] IBM Rational Staff. Ibm rational rose realtime: A guide for evaluation and review, capsules and ports. http://www.ibm.com/developerworks/rational/library/622.html, 2009-02-26.

[17] NCH Software. Express scribe transcription playback software. http://www.nch.com.au/scribe/, 07.05.2009.

[18] Richard Wettel and Michele Lanza. Code city - 3d visualization of large-scale software. http://www.inf.unisi.ch/faculty/lanza/Downloads/Wett2008a.pdf, 2009-05-29.

[19] Robert K. Yin. *Case Study Research. Design and Methods*. Sage Publications, third edition, 2002.

[20] Helen Sharp Yvonne Rogers and Jenny Preece. *Interaction Design: beyond human-computer interaction*. John Wiley and Sons Ltd, second edition, 2007.