An Approach to Visualise and Reconcile Use Case Descriptions from Multiple Viewpoints

Debbie Richards, Anne-Britt Fure and Oscar Aguilera
Macquarie University, Sydney, Australia
Department of Computing
E-mail: richards@ics.mq.edu.au

Abstract

The Unified Modelling Language (UML) is a visual approach to the analysis and design of object-oriented computer systems. However, when it comes to specifying requirements through use case descriptions we see two weaknesses in the current approach. The descriptions are textual (the success and acceptance of UML is largely due to its visual nature) and the group process for acquiring descriptions can result in a small number of stakeholders having their views represented. To address these issues the RECOCASE methodology captures use case descriptions from multiple viewpoints and automatically generates a visualisation of the individual and shared viewpoints to assist identification and resolution of conflicts. This paper briefly describes the approach and results from various initial evaluation studies.

1. Introduction

The capture of requirements from multiple viewpoints has been presented as a way to develop a more complete, consistent and representative set of requirements. [1, 2]. Our approach, RECOCASE, assists the capture and RECOnciliation of functional requirements in the form of use case descriptions from multiple viewpoints. In keeping with the visual nature of other parts of the Unified Modelling Language (UML) object-oriented system development approach, we model the use case descriptions in a diagram to identify and reconcile differences. The RECOCASE methodology includes use case description guidelines, a controlled language to support natural language translation, a requirements engineering process model, a group decision support approach and a Computer Aided Software Engineering (CASE) tool. This paper introduces our methodology, tool and results from initial evaluation studies.

1.1. The RECOCASE methodology and tool

The RECOCASE viewpoint methodology includes the following six iterative phases:

1. Requirements acquisition The first phase begins with the project group, led by a group facilitator (GF), brainstorming a use case diagram. Viewpoints of the use cases are identified and a representative for each viewpoint is appointed. Each representative enters a use case description into the RECOCASE tool.

2. Requirements translation The tool automatically parses the sentences of the use case descriptions into noun and verb phrases by using natural language techniques.

3. Concept generation We novelly apply Formal Concept Analysis (FCA) [3] to generate concepts based on the words and phrases in the use case description. The concepts are ordered using the subsumption relation ≥ and visually displayed as a line or *Hasse* diagram (Fig. 1). The parts of the sentences are shown at each node. Nodes belonging to the same sentence are connected with lines.

4. Concept comparison and conflict detection The GF creates the diagrams by selecting the sentences he/she wants to be included. Figure 1 shows a diagram for all sentences in the "Rent Video" use case concerning the term "rental card". The group members and the GF can use the line diagram to compare viewpoints and detect conflicts. If a viewpoint agent has written a sentence using concepts unlike any of the other group members, this sentence will not share nodes with any other sentences. Use case sentences which have similar concepts but have been expressed differently or at different levels of detail, will share some nodes.

5. Negotiation and Reconciliation During the reconciliation process, the group and GF will review each sentence in sentence number order. Sentences to be included are added to the final use case description. If none of the participants want the sentence, it is marked 'Ignore'. If they do not agree or are not sure, the GF marks it 'Delay'. If the group encounters a sentence that they think has been covered already, it can be marked with 'Done'. Finally, the GF and the test participants view the shared use case and make any desired modifications.

6. Evaluation We use graph theory on the lattices to evaluate if the different viewpoints have become more similar and if a new round of negotiation is necessary.

1.2 Evaluation studies and results

From a number of previous studies involving a total of around 450 second and third year computing students over the past two years we have found that students prefer to use the line diagram for comparing requirements and that they were able to do so more quickly and accurately than with comparisons of textual descriptions. In early 2002, we conducted a study with 201 2nd year students. The focus of that study was to evaluate the guidelines and the learnability and comprehensibility of the line diagrams. We found that test participants with guidelines were more likely to use the same word to refer to the same object and avoid the use of pronouns, modal verbs, adverbs, conjunctions and disjunctions. We also discovered that answers were more likely to be correct when using the diagram as opposed to textual sentences, and that most of the test participants preferred to use the line diagram instead of sentences.

With such promising results we wanted to test whether the group process we had developed was practical, would produce a shared set of requirements and whether that set was more representative and comprehensive than would have been achieved through the normal process of defining use case descriptions. We conducted three separate studies using the RECOCASE approach involving a GF and three students in each group. In a fourth study the group worked together to develop a use case description without the assistance of RECOCASE. Our results show that each final and shared use case description was longer than the individual use case descriptions. All participants were satisfied with the process and the outcome and were able to create a shared use case description using the technique. The most striking difference was found in the attitudes and cohesion of the groups that used the process compared to the group that did not.

1.3 Conclusion

The UML is a visual approach to the analysis and design of object-oriented computer systems. To capture requirements a use case diagram is used to gain a high level view of the main chunks of functionality. However, while agreement at this high level may be achieved relatively easily, it is not until the use cases are described that the stakeholders have any idea if what they have conceptualised is similar to the other stakeholders. The steps that one stakeholder might include in one use case may be different and/or distributed over a number of use cases by another stakeholder. It is therefore very important that use case descriptions are adequately proposed, analysed and agreed upon by the group. The visual nature of the UML is partly responsible for its success and acceptance. The fact that UML only offers visualisation of use cases at a very high level is thus a shortcoming. Our visualisation of use case descriptions is offered as a solution. Further gaining a shared view of the requirements is not well-supported by the standard approach to use case description acquisition. The RECOCASE approach specifically addresses the issues of capturing all the key viewpoints, encouraging participation and identification and reconciliation of conflicts between viewpoints. We believe a set of use case descriptions owned and defended by the stakeholders will provide a better basis for system design and increase the likelihood of acceptance of the delivered system.

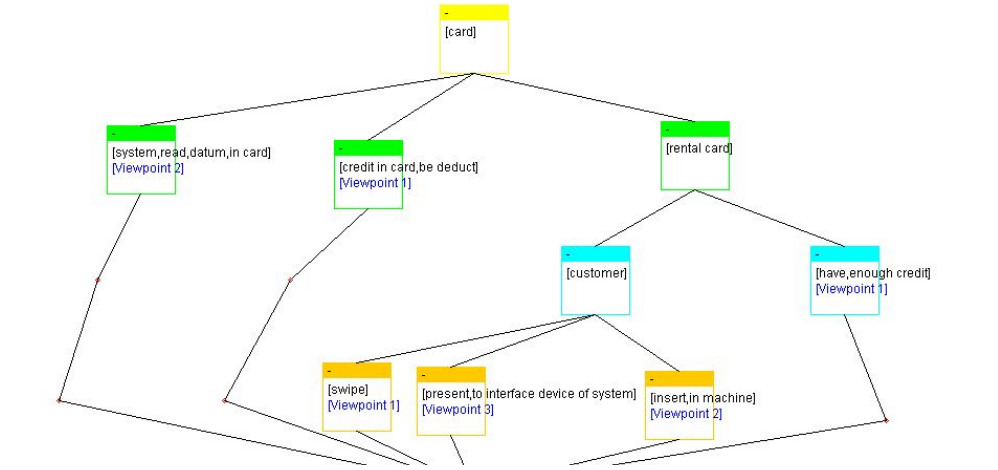

Fig 1. A line diagram from the Group 1 usability study. To read, start with the top node and pick up the parts of the sentence by following all descending pathways. The node at the end of each pathway identifies the viewpoint owner. A number of differences/ similarities can be seen. For example, following the paths that lead down to the lowest nodes, we can see that Viewpoint1 thinks the card should be "swiped", Viewpoint3 thinks the card should be "presented" and Viewpoint2 sees that the card should be "inserted". The group will need to decide whether to commit to a particular type of interface device or to leave the requirement at the more abstract analysis level as Viewpoint 3 has done.

References:

[1] Easterbrook, S. (1991) *Elicitation of Requirements from Multiple Perspectives PhD Thesis*, Dept. of Computing, Imperial College of Science, Technology and Medicine, University of London, London

[2] Ramesh, B. & Dhar, V. (1992) Supporting Systems Development by Capturing Deliberations During Requirements Engineering, *IEEE Transactions on Software Engineering* , 18(6), pp. 498-510

[3] Wille, R. (1992) Concept Lattices and Conceptual Knowledge Systems *Computers Math. Applic. (23)* 6-9: 493-515.