There is a strong need for diff/merge tools on models

Pär Emanuelsson
Ericsson AB
Linköping
Sweden
+46 730436147
par.emanuelsson@ericsson.com

ABSTRACT

We explain why we think there is a strong need for diff/merge tools on models. Then we summarize our previous work in the area.

Categories and Subject Descriptors

D.2.7 [SOFTWARE ENGINEERING]: Distribution, Maintenance, and Enhancement – versioning, difference, diff/merge.

General Terms

Management, Design, Verification.

Keywords

Versioning, difference, diff/merge, model merge, parallel work, team coordination, industrial experience.

1. INTRODUCTION

The greatest obstacle for widespread and successful use of modeling of large applications is the lack of good and well functioning tools for versioning, diff and merge. This awareness has gradually emerged during the last years within Ericsson, who is one of the major users of commercial modeling applications. Most presentations on modeling that we see internally contain a list of remaining problems to be solved and on the top of that list is mostly “merge of models”.

2. GROWING NEED

We also see that this need is growing and is not likely to decrease for some years to come. The traditional way of organizing development has been to have separate large departments that take responsibility for requirements, system architecture, development and test respectively. The current way is to organize in an agile way, that is to establish multidisciplinary teams. Each of these teams have a number of features that they take the whole responsibility for and handle requirements, architecture, design and test for the whole set of features within the team. Hence we have a change from developers that have responsibility for a number of software units to developers that have responsibility for a feature that spans over many software units. The work will be more parallell and many people will be doing changes on a given software unit. It will be harder to try to avoid doing parallell changes.

3. AVODING MERGE

Currently many modeling users have tried to avoid merges as much as possible. The basic technique has been to increase detailed planning in order to avoid parallell work on single software units as much as possible. Of course this decreases efficiency and puts the focus away from producing a consistent and fault free product.

4. MERGE TOOLS IMPOSSIBLE?

Many software people believe that merging tools for models is very hard if not impossible to make and that merge on models is very different from merging text files. It has even gone so far that many think that modeling and agile style of working cannot be combined, “we work agile so we cannot use modeling”.

On the contrary we believe that merging models does not have to be that different and that reasonable merging tools should be not that difficult to design and implement.

5. WHAT IS NEEDED?

Users typically work with large models with thousands of objects and do a few changes to a few objects before checking in, the models to be merged are not made by different projects or different organizations. The users in a project use the same tools and historic versions are available, which makes three way merges possible.

What basically is needed is a tool that can discover syntactic conflicts between changes and display the relevant part of the model and the proposed changes. Then let the user decide which changes to apply and let the user do some editing if desired.

6. USE AND EVALUATIONS

We have used several diff/merge tools since more than 10 years. For UML models we have primarily used IBM/Rational, but also Rhapsody tools. We have also used SDL tools for SDL models. We did an evaluation of Rational diff/merge tools in 2003 and an evaluation of IBM/Rational (RSA/RTx) and Rhapsody tools in 2008. We continue to use IBM/Rational and Rhapsody tools.

We reported on a literature survey of academic research on model merge and an initial analysis of similarities and differences between text merge and model merge [1]. In a later paper, we proposed and discussed the consequences of a number of use cases for text and model merge, based on problems and

Conference'10, Month 1–2, 2010, City, State, Country.

suggestions that emerged from interviews with developers at more sites within Ericsson AB [2]. In [4] we give a more thorough analysis of relevant use cases from [2] with the aim to distil a number of requirements for a practical model merge tool. The 2008 evaluation is available as an Ericsson internal report [3].

7. REFERENCES

[1] Bendix, L., Emanuelsson, P.: *Diff and Merge Support for Model Based Development*, in Proceedings of the International Workshop on Comparison and Versioning of Software Models, Leipzig, Germany, May 17, (2008).

[2] Bendix, L., Emanuelsson, P.: *Collaborative Work with Software Models – Industrial Experience and Requirements*, in proceedings of the Second International Conference on Model Based Systems Engineering – MBSE'09, Haifa, Israel,

[3] Nåls, A., Auvinen, J.: *Model Merge Study*, internal Ericsson Technical Report, April (2009).

[4] Bendix, L., Emanuelsson, P.: *Requirements for Practical Model Merge – an Industrial Perspective*, ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems 2009.

[5]