Requirements-Based Product Line Engineering

Mike Mannion
Glasgow Caledonian University
70 Cowcaddens Road
Glasgow, UK
+44 141 331 3279
m.a.g.mannion@gcal.ac.uk

Hermann Kaindl
Siemens AG Österreich, PSE
Geusaugasse 17
A-1030 Vienna, Austria
+43 51707 43288
hermann.kaindl@siemens.at

ABSTRACT

Reuse and requirements are very important for efficient and successful systems development. This tutorial presents the experiences of requirements reuse using a Method for Requirements Authoring and Management (MRAM). MRAM is a method for establishing and selecting from product line requirements. A product line is a group of products within the same market segment e.g. mobile phones. TRAM (Tool for Requirements Authoring and Management) is a software tool to support MRAM that utilises current proven office technology (MS-Word, MS-Access). The tutorial presents the results of MRAM/TRAM as it has been applied to product line engineering of a real-world application.

Categories and Subject Descriptors

D.2.13. [Reusable Software]—Domain engineering.
D.2.1 [Requirements/Specifications]—Elicitation methods.

General Terms

Design, Documentation, Verification.

Keywords

Requirements, Reuse, Product Line Engineering, Variability.

1. DESCRIPTION

There are many open issues about performing reuse and requirements well, in particular the reuse *of* requirements. This tutorial presents a Method for Requirements Authoring and Management (MRAM) that focuses on systematic requirements reuse, as well as experiences of using it in a real-world application.

For modern, highly complex, high reliability systems, the need for properly structured, carefully controlled requirements specifications, which are understandable, complete and consistent is essential in order for the resultant computer-based system to be delivered on time, within budget and to the desired high level of quality. One approach to managing these problems is to establish a structured pool of reusable product line requirements and to construct the requirements for a new system by making a systematic and guided selection from the pool. A product line is a group of products within the same market segment e.g. mobile phones, pensions or spacecraft control operating systems. An issue with such an approach is the efficient and clean selection of a valid combination of requirements. A valid combination is one in which the requirements selected satisfy any constraints imposed by the product line model.

A product line model consists of a structured pool of numbered, atomic, natural-language requirements, a domain dictionary and a set of *discriminants*. A discriminant is any requirement which differentiates one system from another. Discriminants and parameters represent *variability* in this approach. The model contains all requirements in all the existing systems in the product family and is constructed as a lattice of parent-child relationships. The requirements in the model are all related to each other in parent-child relationships or through their relationship with the discriminants. A discriminant is the root of a tree in the lattice. The nodes below the root are the variants. Requirements belonging to each variant appear in the tree beneath the variant.

In building single systems the first task is to select desired requirements from the set of product line requirements. Existing methods do not provide process details for this task. In our method making choices at discriminant points drives selection from the model. Requirements belonging to the chosen variant appear in the single system model. This discriminant-based method gives superior performance than free selection.

Using MRAM means the management of the requirements definition process is more effective and efficient, producing more accurate and complete requirements documents. TRAM (Tool for Requirements Authoring and Management) is a software tool to support MRAM that utilises current proven office technology (MS-Word, MS-Access). The tutorial presents MRAM and TRAM as well as the results of applying MRAM/TRAM to product line engineering of a real-world application.

The tutorial is aimed at practitioners and academics who want to achieve significant reuse and have some knowledge of requirements engineering, component identification and the problems of developing medium to large computer-based systems. Sufficient introduction will be provided about the real-world application for the audience to understand the principles of product line engineering. After this tutorial the participant will be able to:

- describe a product line engineering lifecycle;
- build a product line model;

ESEC/FSE 2001, Vienna, Austria
© ACM 2001 1-58113-390-1/01/09…$5.00

- develop a single system model from a product line model.

The technical content of the tutorial will include:

- **Product Line Engineering**: Introduction; Motivation; Terminology, Product Line, Product Line Engineering Lifecycles; Benefits.
- **Introduction to Case Study**: Background; Description, Results.
- **Product Line Models**: Domain Dictionary; Reusing Requirements; Product Line Requirements; Requirements Attributes; Structuring Requirements; Requirements Specification; Writing Reusable Requirements; Viewpoints; Discriminant Types; Identifying Discriminants.
- **Building A Single System From a Product Line Model**: Navigation through a Product Line Model; Selecting Requirements; Free/Directed Choice.
- **Tool Support**: A Product Line Tool using Office Products; Requirements Metamodel.
- **Summary**: Review of work; Conclusions; Future Work.

2. REFERENCES

[1] Mannion, M. Lewis, O., Kaindl, H., Montroni, G., Wheadon, J. Representing requirements of generic software in an application family model, Lecture Notes in Computer Science (LNCS 1844), *Software Reuse: Advances in Software Reusability*, (ed W Frakes), 6th Int'l Conference, ICSR6, Vienna, Austria, 27-29 June 2000, ISSN 0302-9743, 153-169.

[2] Kaindl, H. Difficulties in the transition from OO analysis to design, *IEEE Software*, 16, 5, (Sept/Oct 1999), 94-102.

[3] Mannion, M., Keepence, B. Supporting product family variability using patterns, *IEEE Software*, 16, 4, (July 1999), 102-108.

[4] Mannion, M., Keepence, B., Kaindl, H., Wheadon, J. Reusing single system requirements from application family requirements, in *Proceedings of 21st IEEE International Conference on Software Engineering (ICSE'99)*, (Los Angeles, California, 16-22 May 1999), 453-462.

[5] Mannion, M., Harper, D., Keepence, B. Domain requirements definition using viewpoints, *IEEE Software*, 15, 1, (January 1998), 95-102.

[6] Kaindl, H. A practical approach to combining requirements definition and object-oriented analysis, *Annals of Software Engineering*, 3, 1997, 319-343.

[7] Keepence, B., Mannion, M., Smith S. SMARTRe requirements: writing reusable requirements, in *Proceedings of IEEE Symposium on Engineering of Computer-based Systems* (Tucson, Arizona, 2-5 March, 1995), ISBN 0-7803-2531-1, 27–34.

[8] Mannion, M, Keepence, B. SMART requirements, *ACM Software Engineering Notes*, 20, 2, (April 1995), 42-47.