

Empowering Requirements for a Product Family

Elena Pérez-Miñana,
Philips Research Laboratories, Redhill,
Surrey RH1 5HA, UK
elena.perez-minana@philips.com

Paul Krause,
Philips Research Laboratories, Redhill,
Surrey RH1 5HA, UK
& Dept of Computing, University of Surrey
paul.krause@philips.com

Pierre America
Philips Research, Prof. Holstlaan
4, 5656 AA Eindhoven, The
Netherlands
pierre.america@philips.com

Abstract

Document-based management of requirements can be effective for single site, narrowly scoped projects. However, we firmly believe that it is essential to move to a database-centered way of managing requirements if platform-based product development and enterprise wide requirements management is to be fully supported. This paper introduces an approach to the specification of system families that is database centered. It outlines the experience of producing a database of requirements for the TV product. The schema was built using RTM, a requirements management tool supplied by Integrated Chipware. It constitutes a practical example of the advantages that can be gained by changing from the conventional document based requirements management process. It also shows the difficulties that need to be dealt with when mapping from a document-based to a database centered product specification.

1. Introduction

When conceiving new products in the context of a product family it is important to specify *what* is to be expected in a sufficiently precise way so that all stakeholders (from marketing, development, etc.) agree without the possibility of significant misunderstandings. These requirements specification should make clear the properties of each individual member of the family while also making explicit what they have in common.

The main thesis of this paper is that a requirements process for product family development needs to be supported by a move from document centric to database centric management of requirements.

The method on which we have based our study has been developed and applied in Philips [1]:

- It includes the means to arrive at a complete and coherent set of specifications. The use case based approach [5], complemented by a requirement

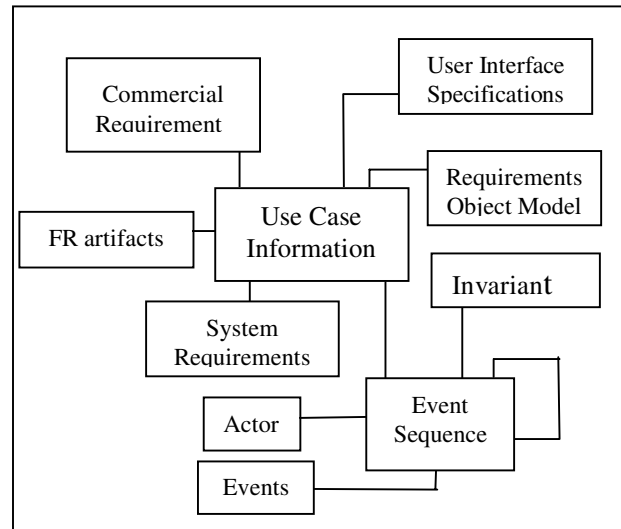


Figure 1 RTM Schema for use with Use-Case based functional requirements

object model, provides a sound basis for eliciting and specifying a product family's requirements.

- It provides a method for refining the use case based requirements into a component model for software development.

There are other methods available but none integrate as well into our work [3].

2. Implementing a database of requirements

The model included in Figure 1 is a simplified version of the RTM schema implemented to store the use case based requirements that specify a TV product family. A successful operational environment must provide effective management of the underlying diversity, homogeneity, and evolution of such specifications. These issues will be discussed in the following points.

As the underlying method employs use cases and a requirements object model to describe the requirements, it was necessary to distribute the solution across two models, i.e. a relational database for storing the use

cases and a class diagram, part of which is shown in Figure 2, for the requirements object model.

In the use cases of the document-based specifications, diversity is handled at two levels of granularity. Each use case has a Region attribute, denoting to which region(s) it is applicable. Additionally, each of the events specified in the flow of events of a use case has a "diversity" attribute associated to its position in the event sequence. In RTM we have handled diversity in two ways: through specific attributes associated to the schema's entities, and by using logical groupings of entities called "collections". For example, all the use cases that are applicable to Europe could either have a Region attribute with "EU" value, or could be non-exclusively linked to an "EU" region collection. Similarly, the diversity of the event sequence can also be handled through "collections" linking the appropriate instances of the "Event Sequence" and "Events" entities. It is still unclear whether both forms of diversity are needed or whether one of the two is sufficient to handle any diversity in any context.

The homogeneity is handled through the "Invariant" entity. They are constraints on the relations between states or attribute values for all members of the product family. The classes and associations of the requirements object model also provide alternative ways to describe homogeneity. For example, an inheritance relation between the AudioProcessing and TrebleBass classes indicates that the Treble and Bass controls are either both present or absent in a particular TV. Homogeneity is also expressed by the constraints on the multiplicity of different associations.

Apart from the issues described, there is the additional difficulty of providing sufficient traceability links that will handle change through time [4]. We must ensure that the model will be flexible enough to manage this evolution. In RTM this is achieved through the "change request" feature. It allows us to keep track of the changes made to any requirement stored in the database, together with a rationale for each particular change.

3. Conclusions

The magnitude of the changes required to handle product family requirements in the present working practices is probably greater than initially envisioned. Nevertheless, the benefits that can be gained in terms of, e.g. product quality, efficient working practices, improved multi-site development, make it worthwhile.

The effort invested in designing a database model for storing requirements is extremely important. It requires skill and patience to get the database schema right.

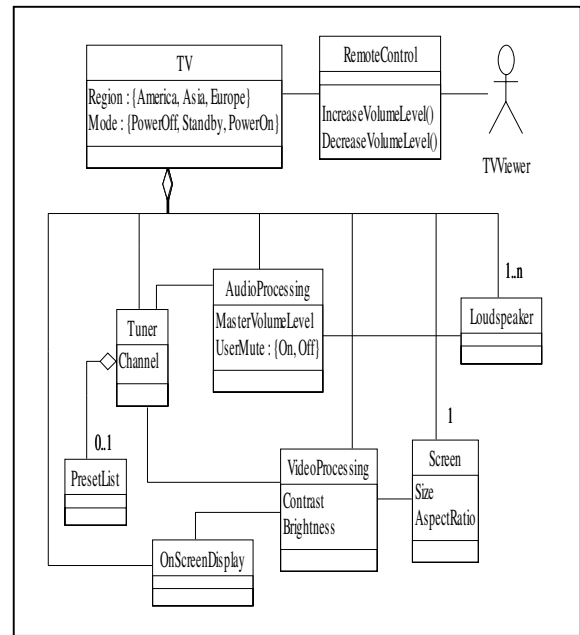


Figure 2 Requirements object model for the TV

Nevertheless, doing so makes the subsequent management of the requirements much easier.

We hope, during the course of this year, to demonstrate the feasibility of a basic traceability process. The database of requirements described is a first step in this direction.

4. Bibliography

- [1] P. America, J. van Wijgerden, *Requirements Modeling for Families of Complex Systems*. Paper at IW-SAPF 3: Third International Workshop on Software Architecture for Product Families, March 15-17, 2000, Las Palmas de Gran Canaria, Spain. Springer LNCS 1951.
- [2] P. America, H. Obbink, R. van Ommering, F. van der Linden, *A Component-Oriented Platform Architecting Method Family for Product Family Engineering*, SPLC-1, First Software Product Line Conference, Denver, Colorado, August 28-31, 2000.
- [3] C. Atkinson, J. Bayer, C. Bunse, E. Kamsties, O. Latinberger, R. Laqua, D. Muthig, B. Paech, J. Wüst, J. Zettel, *Component-based Product Line Engineering with the UML, The Kobra Method*, UML-2000, York, October 2000.
- [4] R. Balasubramaniam, M. Jarke, *Towards Reference Models for Requirements Traceability*, IEEE Transactions on Software Engineering, 27,1,58-93, January 2001.
- [5] A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2000.