Framework for documenting design decisions in product families development

Alejandro Alonso, Gonzalo León, Juan C. Dueñas, Juan A. de la Puente
Universidad Politécnica de Madrid, E.T.S.I. de Telecomunicaciones,
E-28040 Madrid, Spain
e-mail: aalonso@dit.upm.es, gonzalo@dit.upm.es, jcduenas@dit.upm.es, jpuente@dit.upm.es

Abstract

The lack of documentation for describing the development of an architecture is common in many software systems that may cause problems in later developments. This paper presents a framework for representing design decisions in the development of product families. The information is structured as a DDT (Design Decision Tree), where nodes represent design decisions and branches relate nodes. The access to this information is based on HTML and WWW navigators. The use of a simple case study is proposed in order to analyse its feasibility.

1. Introduction

Usually, the decisions taken in the development of a software architecture and their rationale are not explicitly documented. In these cases, the only information available is the architecture definition. Although this information is very useful, it is not enough. This picture is even worse when dealing with product families, i.e. software systems embedded in different versions of a certain industrial product. Although the designs for the different products are strongly related, in most cases there is no documentation on the relations and differences among them and the reasons why they differ. Some potential problems related to this lack of information have been identified in industrial practice:

- Lack of conformance to the architecture of different software products.
- No structure describing all the members of product families, their features and differences.
- Changes that violate architectural constraints.
- Management of global quality attributes.

One way of alliviating these problems is embedding architectural decision making into the software development process. In this way, system designers could access to the rationale of the decisions taken when developing previous products and assessing if this decisions are still valid for the new product. In addition, this information is a way of representing design experience that may be used for the design of other systems.

The goal of this paper is to propose a design decision framework for supporting the documentation of architectural decision making into the software development process. Hence, architectural decisions should be an important output of the design phase, and not only a side effect of the derivation of a system architecture. In this way, the architecture description in conjunction with the decision documentation will serve to alleviate some of the problems stated above.

In the rest of this paper, different aspects of the decision making framework are dealt with. Section 2 analyses the goals and requirements for this framework The information structure to be used is presented in section 3. Some techniques for supporting the decision making process are mentioned in section 4. Section 5 includes the means for accessing this information, while section 6 includes a reference to a case study.

2. Decision Making Framework

The goals for this decision making framework are the following:

- Describe an information structure for supportting the documentation on the decisions taken.
- Provide some guidelines on the techniques to be used to support the decision making.
- Document the decision making process for a system. It will include not only the development but also the maintenance activities. In addition, the framework could be used for the decision making information for a product family.

- Relate the decisions taken with the development of the decisions. For example, if it has been decided to use some specific criteria for partitioning a system into subsystems, the framework could reference the results of this partitioning.

2.1. Decision making framework requirements

The following requirements have been found necessary for representing the information related to the decision making:

- Multi-perspective.
- Visual representation.
- Complexity control.
- Group facilities.
- Gradually formalisable.
- Scalability.
- Compatibility.

In addition, the problems detected in current industrial practices should be taken into account in the design of this framework. In particular, the decision making framework may also be useful for dealing with the following issues:

- Relate component changes throughout the product architecture and/or product family.
- Reflect diversity and complexity in software architecture.
- Document generic of specific parts of the software architecture.
- Describe an architecture of a family of systems and derive new family members from existing ones.

3. Information structure

First of all, we are going to deal with the information structure for handling all the decision making process. For this purpose, the Design Decision Tree (DDT) [4] is a suitable basis, although it will be necessary to improve it to deal with some of the previously stated goals and requirements. DDTs were designed for incrementally specifying and refining the architectural knowledge for a system. Each node is used for documenting a design decision In this paper, it is proposed that the information that a node should have is the following:

- Partial specification of the problem to be handled
- Decision of an approach towards their satisfaction.
- New requirements and constraints imposed by the approach.
- References to further design decisions (nodes) that refine the specification of the problem and the solution

DDT's are suitable for structuring the decision making information on a system development. If we deal with the maintenance phase and product family evolution, the structure will derive to a directed graph. An initial node should exist for each of the products in the family, as they have slightly different requirements. As far as these requirements are similar to those of previous members of the family, certain decisions will be the same, and hence, the nodes that represents them. However, as there are differences in the requirements, additional nodes may appear caused by the different choices that should be made in order to fulfil the requirements. These new additional nodes represent the variation in the product family.

With respect to the structure of the node, the information proposed in [4] needs to be incremented. First, it is interesting to link a decision with its consequences. This is specially clear when a decision causes a refinement in the design of the system. For example, a certain decision may lead to a specific partition of the system into subsystems. As a consequence, it is useful to relate the decision node in the DDT with the design documentation related with that design step.

In addition, the information on the evaluation of different alternatives may be complex, and hence it may be too large to be included in a single node. On the other hand, it is useful to keep it, as partially documenting by itself the decisions taken, and as a possible eligible alternative if the requirements change due to maintenance or to derivation of a new product. Hence, an additional kind of node are envisaged.

In summary, the information structure we propose in the decision making framework is based on DDT, with the following improvements:

- DDT structure for holding information of decisions dealing with maintenance and product families.
- Each node may have links to the documentation of the design actions caused by the associated decision.
- Each node may have links to the documentation on the evaluation of different alternatives.

4. Techniques for supporting the decision making

In this section, simple guidelines on some candidate techniques that can be used to support the decision making process are provided. As commented previously, the preferred technique depends on the type of decision it is needed to make and the abstraction level of the system information available. If we have to decide on the achievement of an architectural description with respect to certain quality attributes,

then the specific assessment techniques could be used.

If the decisions to be made are related with the initial design steps, when the material available are basically a set of requirements, then the assessment techniques are not useful, because there are no architectural description. Then different techniques could be used:

- Ishikawa diagrams [3] are useful for early brainstorming sessions, intended to describe the features of the products and potential requirements (these phases may be outside of the scope of the decision framework).
- QFD (Quality Function Deployment) and RFD (Risk Function Deployment) [3] are useful to prioritise alternatives and correlate and estimate the associated risks.
- Finally, there are several sources of knowledge that can be used to support the decision making, such as previous experience, design methodologies, design patterns, and development techniques for specific application domains.

In the decision nodes should be included links to the most relevant sources of information, in order to have a picture of the decision process as complete as possible.

5. Accessing the information

A useful basis for accessing the information in theesign decision framework is an intranet architectural WWW server and documents on HTML. The main advantage is that clients (WWW navigators) are widely available and the cost is very low compared with other approaches. On the other hand, many implicit concepts like navigation, hyperlinks or plug-in of tools have a direct interest in our case.

One additional advantage is that it does not assume the existence of a single repository of information. It is a responsibility of the designer (probably with some hints embedded in the system information) to navigate from one server to another. As a consequence, a high degree of flexibility is reached.

6. Case study

In order to evaluate this framework and have a first impression on its viability and usefulness, the decision making information developed for the a teleoperation system [1], will be structured according to this proposal. The small size of this system and the previous work performed allow to have a first prototype in a few weeks.

The results of this case study will allow us to enhance the current proposal for the framework and to use it for documenting the decision making process for an industrial case study.

7. Conclusions

In this paper, a design decision making framework has been proposed. Its main goal is to support the design decision making process in the development of product families. This framework is structured as a DDT. In this way, the information of the design decisions in a product family can be related. This information is accessed based on the use of WWW navigators and HTML.

The results of the application of the case study has been successful. The framework has allowed to organize the design decision information and to ease the access to this. Future work will be based on the application of the framework to an industrial and more complex case study.

8. References

[1] A. Alonso, B. Álvarez, J.A. Pastor, J.A. de la Puente, A.Iborra, “Software architecture for a robot teleoperation system“, *International Workshop on Development and Evolution of Software Architectures for Product Families*, Las Navas, Spain,1996.

[2] J.C Dueñas, M. Hanswirth, “Creating and Using Hyperlinked Software Architectures“, *International Workshop on Development and Evolution of Software Architectures for Product Families*, Las Navas, Spain,1996.

[3] K. Ozeki, T. Asaka, *IHandbook of quality tools. The Japanese approach*, Productivity press, 1990.

[4] A.Ran, J.Kuusela, “Design Decision Trees“, *Proceedings of Eight International Workshop on Software Specification and Design*, Schloss Velen Germany, 1996, IEEE Computer Society Press, 1996, pp. 172-175

9. Acknowledgements

The work presented in this paper is partially supported by the European Communities under the ESPRIT programme, project no. 20477.

The authors wish to acknowledge the participation of the other members of the consortium in the reported work. The ARES (*Architectural Reasoning for Embedded System*) consortium comprises: Nokia Research Center (Finland), Philips Research Center (The Netherlands), ABB Norway (Norway), Imperial College London (United Kingdom), Technical University Vienna (Austria), Technical University of Madrid (Spain).