Attribute-Based Evolution Patterns for Product Lines

Davor Svetinovic and Michael Godfrey
Software Architecture Group
University of Waterloo
Waterloo, Ontario, N2L 3G1, CANADA
Tel. (519) 888-4567 x5388
{dsvetinovic, migod}@uwaterloo.ca

ABSTRACT
Several recent achievements in software architecture and requirements engineering are based on the study and use of quality attributes; these include goal-driven requirements engineering processes [1, 14], Architecture Tradeoff Analysis Method (ATAM) [7], and Attribute-Based Architectural Styles [13]. This paper presents an initial study of the evolution of the systems in terms of quality attributes. The purpose of the study is to put the knowledge about evolution in the form directly usable by other quality attribute based techniques. The paper emphasizes the use of the attribute-based evolution patterns in the reconstruction and design of product line architectures.

Keywords
Software Evolution, Software Architecture, Product Lines, Architecture Tradeoff Analysis Method, Goal-Oriented Requirements Engineering, Architecture Driven Design

1 INTRODUCTION

Previous work in software evolution [4, 5, 8, 10, 11, 12], has shown that software very often evolves differently than one plans and expects. In order to deal with the problems that arise due to this unpredictability, one has to find a way to capture and reuse knowledge about these uncontrolled changes and what causes them. The goal of our current study is to build a systematic process and framework for preserving and applying the evolution knowledge in the attribute-based techniques in the other software engineering areas (for example, goal-driven requirements engineering processes, Architecture Tradeoff Analysis Method (ATAM)). The framework will extend existing attribute theory, and facilitate even more the unification and transitions among different techniques.

The first goal of this paper is to present an initial study in attribute-based software evolution, and the second is to describe the use of the obtained results in the reconstruction and design of product line architectures. Product line architecture supports a wide range of different sets of goals that are specific to different products. These goals are often conflicting and change over time. The study of how these goals change facilitates the design process and improves the quality of the product line.

The study and use of quality attributes has made many contributions to software engineering practice. The goal-oriented requirement engineering processes have helped capture a wider range of requirements then previously possible, improve the requirements traceability, and facilitate the process in general [14]. The Attribute-Based Architecture Styles (ABAS) have allowed qualitative reasoning about the use of a particular architectural style [13]. Architecture Tradeoff Analysis (ATA) method relies upon the use of quality attributes to analyze and express the architectural tradeoffs [7]. The quality attributes are the initial artifact for several architecture design processes, including the Architecture-Based Design Method [1]. Besides providing the driving force, quality attributes also serve as the connection among all these techniques and methods[6].

All of the previously mentioned techniques are based upon a solid understanding of the interdependencies among attributes, especially conflicting ones. Several studies have tackled the problems of complex dependencies between the attributes and how to manage them [2, 9].

In order to optimize the qualities of the system, it is not enough to consider only the quality attributes and their priorities at one moment in time. The qualities of the system change. Also, the stakeholders' quality priorities change. We need a way to prioritize and estimate changes in quality attributes. The systematic study of these changes, and how and why they have occurred will help us achieve this.

This estimations and prioritization is especially difficult in the case of product lines. This difficulty arises from very essence of the product lines, that is, the fact that the core product line architecture is designed without even knowing all the goals of the products it will support. Therefore, the problem is to forecast as closely as possible which qualities the core of the product line will have to support in the future.

The approach that we are taking to solve the problem is to find an initial set of quality attributes, relationships among them, forces that act upon them, and to capture changes that frequently occur in the pattern format. Several evolution case studies for different product families are conducted during which the theory and format are refined. The goal is to obtain an efficient reconstruction process, attribute theory, and format that will allow routine study and documentation of evolution and efficient application of the results in other areas.

In the next section we present the initial attribute theory and pattern formats. In the third section, we explain the use of the evolution patterns in the product line architecture design. We conclude with the discussion of the future work.

2 Quality Attributes and Patterns

As observed previously, the evolution and architecture of a system are very often different from the way we plan them [3, 5]. Many forces act on the development of the system and we need to understand them in order to keep the development under the control. We are using the main groups of these forces to classify the quality attributes, (Figure 1). The main purpose of this division is to identify the major conflicts among quality goals. This division is also useful for the evolution pattern categorization, as we will see later.

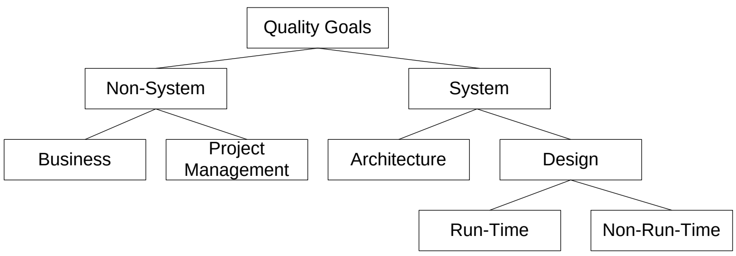

Figure 1: Goal Classification

The business goals are the primary driving force for the development of every system. They are usually very high level goals, such as quick time to market. They are achieved through the project management and architecture goals. Project management goals, beside including positive goals like maximizing developers productivity, include also individual goals. However, these often contradict general management and architecture goals, and are often not taken into account by managers and architects. Architecture goals are the high level system goals, which are often decomposed to and achieved through design goals. The general relationships among groups of goals are depicted in Figure 2.

One particular goal can exist in several groups simultaneously. The problem arises since that goal can have different priority among groups. Even different stakeholders that influence one goal group can give the goal different priority, so the differences arise even within the group. For example, users and buyers have business goals, but users can emphasize more usability over cost. These priorities change over time. New goals emerge over time. This augments the size

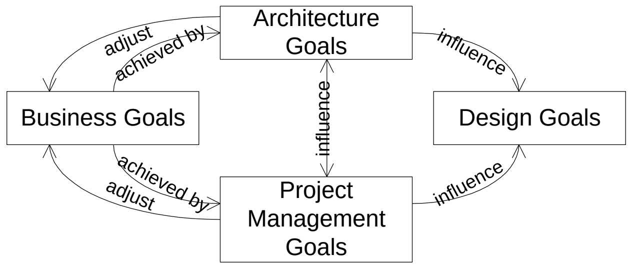

Figure 2: Main Goal Group Relationships

of already existing problem — it is very hard to optimally support all goals and satisfy all stakeholders. A solution is to study common changes that have occurred in already existing product families.

It is useful to make the distinction between the evolutionary changes that are common to all the software systems, and those that are common to a particular product family. In our work, we are working with *evolution stories* to capture the evolution of a product, and *evolution patterns* [5] to capture changes common to many systems. *Attribute-based evolution stories* and *attribute-based evolution patterns* are the subsets of previously mentioned that emphasize the changes in terms of quality attributes. The scope of stories is similar to the scope of reference architectures. It emphasizes the changes particular to the product family and has wider scope than patterns. The scope of patterns is similar to the scope of the architecture styles. Patterns capture common changes that occur in most systems.

In our work, we are concentrating on three quality attribute groups: business, project management, and architecture. The focus of our current study is on documenting the evolution stories for several product families. We will use the results obtained for the stories to produce a set of the patterns observed.

We use the following format for the attribute-based evolution stories:

- *Product Name and Characteristics*
- *Product Family and Characteristics*
- *Development Process* — describes the characteristics of the development process used, number of developers, tools used, etc.
- *Target Market* — the description of the product market including types of users, market characteristics, etc.
- *Stakeholders and Attributes* — captures the relationships among stakeholders and attributes.
- *Measurement Method* — listing of considered attributes, time intervals when observations are taken — most often a stable release, etc.

- *Business Quality Attributes* — During every measurement, for every business quality attribute we capture which attributes support it, which attributes act as an obstacle, for which attributes it acts as an obstacle and which ones it supports. The priority of the attribute is estimated compared to the other attributes. The techniques for achieving that quality attribute are described. Graphs are often used to represent results.

- *Project Management Quality Attributes — same as previous.*

- *Architecture Quality Attributes — same as previous.*

- *Conflicts* — describes the main conflicts that have occurred over time and their resolution.

The format for the patterns is the same except that it is not product specific. The format will evolve and the attribute set will be fixed, as more and more existing systems are analyzed.

In our approach, the first step in the reconstruction of the evolution of the product is to recover the architecture of the system. We use the Portable Bookshelf (PBS) tool [15], see Figure 3, to recover and visualize the architecture of the system. Product documentation is studied in order to recover the conceptual architecture of the system and gather the initial set of goals. Next, the architecture of all the product releases is recovered. The documentation is studied and, if possible, interviews are conducted in order to recover the intentional goal changes. Finally, the recovered architectures are analyzed in order to recover the actual goal evolution. For the last step, we use the intentional goal changes as the reference evolution model, and use concepts of Architecture Tradeoff Analysis Method (ATAM) and Attribute-Based Architectural Styles (ABAS) to recover from the extracted architectures which goals have actually been satisfied. Figure 4 summarizes the approach.

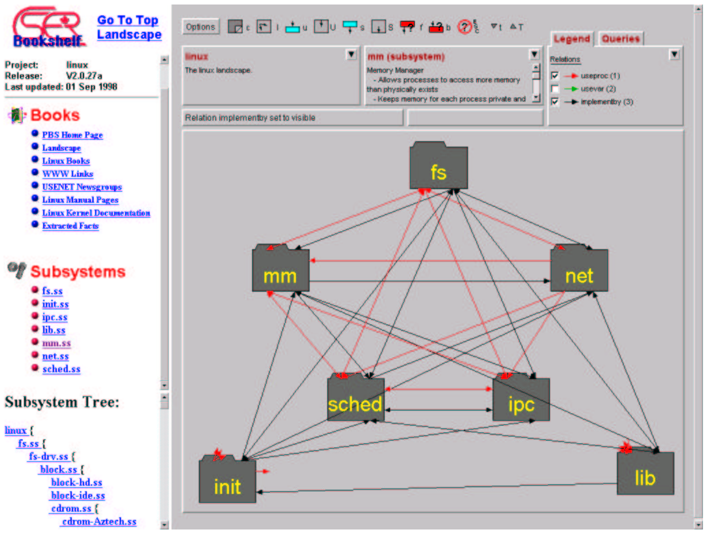

Figure 3: PBS: Architecture of Linux Operating System

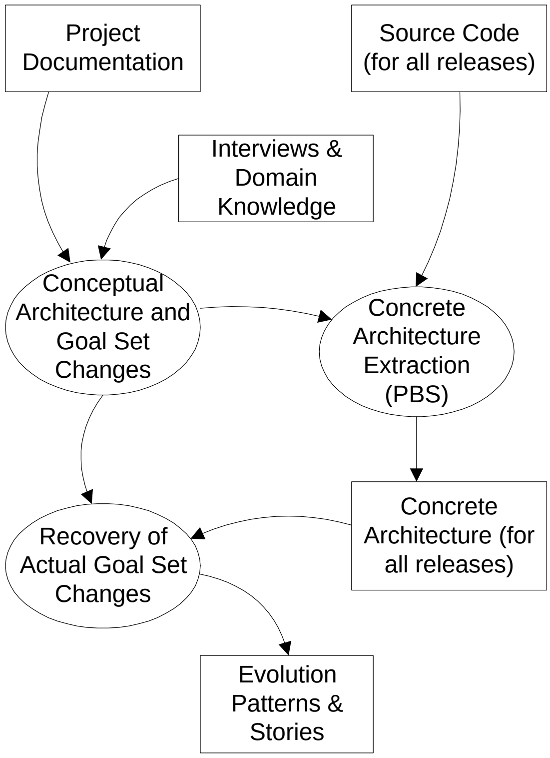

Figure 4: Attribute Evolution Reconstruction Approach

3 Product Line Evolution

In the terms of the attribute-based evolution, product lines can be viewed as the architectural and management technique to satisfy a new set of business and project management goals. The difficulty of converting several existing products into a product line is simply due to the fact that these products are often built without consideration of the set of goals that are accomplished by the introduction of product line.

The reconstruction of existing products and the formation of the product line is the process of identifying commonalities and variabilities among these products. The commonalities can be viewed as the set of common functional and quality attributes. These attributes are supposed to support the variabilities of all the existing products plus the envisioned ones. The same way, product line goals have made the task of unifying the separate products into the product line difficult, the goals of envisioned product can be hard to achieve if not taken into the account during the initial product line construction.

To reduce the difficulty of this problem, the following process can be used:

- Represent every product as a set of functional and quality attributes.

- Find the intersection of these sets. The architecture that supports these goals represents the core of the product line.
- Use the evolution patterns to estimate the conflicts that can arise from the need to support the future goals of the already existing products.
- Refine the core architecture to accommodate possible future goals of already existing products.
- Use the evolution stories for the products similar to envisioned products to estimate possible future conflicts. Also, use evolution patterns to estimate possible future goals based on the already existing set of goals.
- Decide if it will be feasible to support envisioned goals using the proposed product line, and if it will then refactor the architecture of the common assets to form the core of the product line and to optimize the common attributes.

The core idea of the process lies in the optimization of commonalities that are expressed in terms of functional and quality attributes, and in the estimation of future quality attributes. Evolution stories and patterns are basic tool to perform such analysis — they provide codified knowledge based on which one can forecast which goals will have to be supported by the core product line architecture.

4 Conclusion

We have presented core concepts of the attribute based evolution research that is currently in progress. It has the potential of contributing to and improving several state-of-the-art methodologies. It is particulary useful in the area of reconstruction, construction, and forecasting of product lines.

We are currently analyzing the evolution of Linux operating system.

5 Future Work

The future research directions include attribute-based evolution recovery of several systems, development of the efficient process and tools in order to automatize evolution recovery, and the integration of the developed theory with other attribute-based techniques. We will also use stories and patterns to analyze appropriateness of decisions for several already existing product lines.

REFERENCES

[1] F. Bachmann, L. Bass, G. Chastek, P. Donohoe, and F. Peruzzi. The Architecture Based Design Method. CMU/SEI-2000-TR-001.

[2] B. Boehm, and H. In. Identifying Quality-Requirement Conflicts. *IEEE Software*, Vol. 13, No. 2, (March 1996).

[3] I. T. Bowman, R. C. Holt, and N. V. Brewster. Linux as a Case Study: Its Extracted Software Architecture. *ICSE '99: International Conference on Software Engineering*, Los Angeles,California, May 1999.

[4] H. Gall, M. Jazayeri, R. Kloesch, and G. Trausmuth. Software evolution observations based on product release history. *Proc. of the 1997 Intl. Conf. on Software Maintenance (ICSM97)*, Bari, Italy, (October 1997).

[5] M. W. Godfrey, and Q. Tu. Evolution in Open Source Software: A Case Study. *Proc. of the 2000 Intl. Conference on Software Maintenance*, San Jose, California, (October 2000).

[6] D. Gross, and E. Yu. From Non-Functional Requirements to Design Through Patterns. <http://www.hkkk.fi/ mrossi/refsq/f113.pdf>

[7] R. Kazman, M. Klein, and P. Clements. ATAM: Method for Architecture Evaluation. CMU/SEI-2000-TR-004.

[8] C. F. Kemerer and S. Slaughter. An empirical approach to studying software evolution. *IEEE Trans. on Software Engineering*, 25(4), (July/August 1999).

[9] A. van Lamsweerde, and E. Letier. Handling Obstacles in Goal-Oriented Requirements Engineering. *IEEE Transactions on Software Engineering*, Vol. 26, No. 10, (October 2000).

[10] M. M. Lehman and L. A. Belady. Program Evolution: Processes of Software Change. Academic Press, 1985.

[11] M. M. Lehman, D. E. Perry, and J. F. Ramil. Implications of evolution metrics on software maintenance *Proc. of the 1998 Intl. Conf. on Software Maintenance (ICSM98)*, Bethesda, Maryland, (November 1998).

[12] M. M. Lehman, J. F. Ramil, P. D. Wernick, D. E. Perry, and W. M. Turski. Metrics and laws of software evolution - the nineties view. *Proc. of the Fourth Intl. Software Metrics Symposium (Metrics97)*, Albuquerque, NM, 1997.

[13] M. H. Klein, R. Kazman, L. Bass, J. Carriere, M. Barbacci, and H. Lipson. Attribute-Based Architecture Styles. *Proceedings of the First Working IFIP Conference on Software Architecture (WICSA1)*, San Antonio, TX, 225-243, (February 1999).

[14] J. Mylopoulos, L. Cheung, and E. Yu. From Object-Oriented to Goal-Oriented Requirements Analysis. *Communications of ACM*, Vol. 42, No. 1, (January 1999).

[15] PBS: The Portable Bookshelf. <http://swag.uwaterloo.ca/pbs/>.