Evolution of Product Line Architectures

Marcelo Paulo Amaolo

Departamento de Ciencias de la Computación
Universidad Nacional del Comahue
Buenos Aires 1400, Neuquén, Argentina
Phone: (54) 299 - 4490312, fax: (54) 299 4490313
E-mail: mamaolo@uncoma.edu.ar

Abstract: Product-line architectures, i.e. a software architecture and component set shared by a family of products, represents a promising approach to achieving reuse of software. Several companies are initiating or have recently adopted a product-line architecture. However, little experience is available with respect to the evolution of the products, the software components and the software architecture. Due to the higher level of interdependency between the various software assets, software evolution is a more complex process. We identified characterization of software product lines based on dimensions of primary assets, views on the organization and on assets life cycle stages and after that introduced categorizations of the evolution of the requirements, the software architecture and the software components. Our work is focused on analyzing different ways of managing modifications during architecture evolution.

1 Introduction

A great number of organizations have realized that the traditional software engineering approach, otherwise known as single-product approach, cannot guarantee the speed needed to introduce new functionality to their released systems as well as new products to satisfy customers requirements. [Bas98, Bas97, Gre02, Cle01]. These organizations have explicit needs to achieve large-scale productivity gains, improve time to market, maintain market presence, compensate for an inability to hire, and leverage existing resources [Cle01]. Some of them are finding that adopting approaches that emphasize proactive reuse, interchangeable components and the practice of building sets of related systems together can yield remarkable quantitative improvements in productivity, time to market, product quality, and customer satisfaction. [Cle01, Gre02]. These methods, referred to as *software product line* or *software family practices* have developed around these approaches.

A *software product line* is a set of software intensive systems sharing a common, managed set of features that satisfy the needs of a particular market segment or mission and that are developed from a common set of assets in a prescribed way [Cle01].

We define a software asset as a description of a partial solutions or knowledge engineers use to build or modify software products [Bas99]. A software architecture is a structure or structures of the system that consist of software components, the externally visible properties of those components and the relationships among them [Bas99, GS96].

It is most economical to build a software product line as a product family, where a product family is a group of systems built from a common set of assets. In fact, the products in a software product line can best be leveraged when they share a common architecture that is used to structure components from which the products are built [Cle01, Lin02]. This common software architecture capitalizes on commonalities in the implementation of the line of products and provides the

structural robustness that makes the derivation of software products from software assets economically viable. The architecture and components are central to the set of core assets used to construct and evolve the products in the product line. When we refer to a product line, we always mean a software product line built as a product family [Bas98].

By product line practice, we mean the systematic use of software assets to assemble, instantiate, generate, or modify the multiple products that constitute a product line [Jon02]. Product line practice involves strategic, large-grained reuse as a business enabler, and as organizations experienced considerable savings in using a product line approach, other organizations are attracted to the idea but are in varying stages of integrating product line practices [Cle01].

Organizations have recognized the importance of maintaining a product line architecture when moving from the single product at-a time to a development model in which a family of products is developed in a coordinated fashion. But much of the research efforts today regarding product line architectures is directed towards the initiation of a product line, and as a consequence, its evolution is not as well studied [Sva00, vdH01].

In section 2 we describe a characterization for software product lines based on the decomposition in three perspective of analysis which help us identified the field of study. In section 3 we briefly introduce the problem of evolution and a categorization of software product line evolution. We discuss future work and conclusion afterwards.

2 Decomposing software product lines

We can describe three dimensions in which the concepts included in software product lines can be decomposed [Bos00]. The first dimension is divided according the *primary assets* that are part of the reuse-based development. In this dimension the product line in described in terms of the *software architecture*, the main asset of the product line. The main activity is to design the architecture for the product line that covers all the products in the line and include the features that are shared between products (commonalities) as well as theirs differences. The seconds set of assets are the *components* that have been identified in the product line architecture, and has to be specified their functionality, variability and requirements. The final set of assets are the *systems* constructed based on the product line architecture and its components. This activity requires the adaptation of the product line architecture to fit the system architecture, which may require the removal and adding of components to the architecture, the adding or removal of relations between components, the configuring of the components for the system and the development of system-specific code.

The second dimension to which one can decomposed the product line architecture is concerned to the various *views* on the organization. The SEI workshops on PLA [Bas 97, Cle01, Bas98] use this decomposition to describe issues related to product line architecture. These views include the *business* view, with the important aspects of the considerable investment required to convert from product oriented one at-a-time approach to software development to a reuse oriented product line base approach, and the delay introduced when an organization turn to this paradigm. The *organizational* view is also included, as the conversion to reuse based software development has

organizational effects as well and these need to be addressed. At last, the *process* view and the *technology* view, considering the fact that a product line architecture and a set o reusable components has considerable effects on the processes associated with product or system development, the introduction of product line specific processes, and the technology needed including the use of Object Oriented frameworks, assessment techniques for maintainability and other quality attributes, etc.

Finally, the third dimension described based on the *life-cycle* of each of the assets in the organization. Primary the *development*, concerned with the point at which the main assets are developed initially either from scratch or based on legacy code, introducing several aspects specific to this phase in the life-cycle of these assets. Then the *deployment* or instantiation of the components, as the use of the product line architecture and components does not imply simply coping the shared assets but the adaptation, configuration and extension for their use in the product. And after that all the three main assets *evolve* constantly because of the requirements on the products evolve, so these activities justify the differencing from the first two phases in the life-cycle of assets.

3 Major Issues

In the traditional one at-a-time life cycle software development a sizable amount of money and time is spent due to evolution of software in the maintenance phase. As users requirements change differentiating from initial requirements, the system is normally replaced by subsequent new versions, usually developed by a different group of the one taking care of the maintenance and customer support. [Sva00, Aji00].

In the product line approach the produced software is arranged around commonalities that are shared by a family of products, which are considered in the architecture as one of the software core assets. The product line architecture is designed to support a certain amount of variability in the product level. Those functionalities that aren't supported by the common architecture can be implemented in the product level.

Only when the functionality can be reused in another product, the ones are generalized and because of that becomes a new core asset. Thus, a product line is a continuously evolving mechanism. The dependency between the various assets in the product line is very complex due to the multiple artifacts involved and it may be difficult to maintain the status of assets [Gre02, Cle01].

In the product line approach, released products do not necessarily share a single system version, but can be built on various versions of core assets and glued with product specific code. This makes configuration management and tracking of different version and variations more challenging than in traditional single software development.

New requirements change can affect both system functions as well as business goals on with the product is developed. In the business view, is must be considered if the requirements incorporation affect existing products or if they include the process of launching a new product to be developed. In order to manage the kind of evolution discussed, organizations must define clearly the support evolution process and responsibles [Lin02].

The decision not only change the product but the product line architecture and the product line requirements on the asset set. The effect of the spanning must be well managed to preserve the consistency of the asset base, and the handling of changes must include the effective consideration of impact and propagation, in line with the main goal of product line engineering, which is reduce cost and increase customers satisfaction.

We can distinguish a number of categories regarding evolution in the dimension of the life-cycle of the primary assets: evolution of the requirements, evolution of the product line architecture and of the components in the product line architecture. The changes that affect the product line, from the evolution of the requirements view, can be classified as: [Sva00, Sva99, Har01, Bos00, Pus02]

- Introduction of a new product line, usually based on an existing one
- Introduction of a new product, needed as a demand from the market
- Adding new features, from customers or competitors as an improvement of the functionality
- Extension of standard support, concerned with changing an existing framework implementation to add functionality
- New version of infrastructure, extension of the underlying hardware or the operating system
- improvement of quality attributes, of the product line members and of the assets in the product line.

The requirements of the product line will affect two areas: the architecture as a whole and the components in the architecture, as:

- Split of software product line, when a new product should be developed
- Derived product line architecture, as a sub product line as requirements indicate
- Creation, Modification, Replace or Splitting of product line architecture components
- Creation or Modification of then relations between components

The use of product line architectures provides a particularly rich source of changes: new products are introduced, existing products are enhanced and modified, and old products are retired. Methods exist that record these kinds of changes by maintaining explicit representations of the evolution of a product line architecture. Despite the availability of such representations, it still is difficult to quickly gain an understanding of the exact changes that define the difference between two products [VdW02].

4 Conclusions and Future Work

We presented briefly the considerations that should be applied to a single product or the whole product line when customers require new functionalities and other improvement for a product. The product line approach is a valid strategy to improve organization software production throughput and satisfy customers requirements. [Pus02, Bas98, Cle01]. In the next stage of our work, we will inspect different ways of managing modifications during architecture evolution. The evolution of the product line can be recorded using an architectural description language and with other alternatives strategies [Pus02]. As analyzing guidelines for software product line evolution, we will focus the evolution of a product line system in terms of organizational alternatives, how can be

measured evolution and variability of systems in the development process, and the characteristics of the architecture evolution environment in which we can process this variability handling.

5 References

[Aji00] Ajila, Samuel, "Change management: modeling software product lines evolution". *PDSTD'02 – SCI2002 / ISAS2002*, Orlando, USA, 2002.

[Bas97] Bass, L.; Clements, P.; Cohen, S.; Northrop, L.; and Withey, J. "Product Line Practice Workshop Report" (CMU/SEI-97-TR-003, ADA 327610). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997.

[Bas98] Bass, L.; Clements, P.; and Kazman, R. "Software Architecture in Practice". Reading, MA: Addison-Wesley Longman, Inc., 1998.

[Bas99] Bass, Lenn; Campbell, Grady; Clements, Paul; Northrop, Linda and Smith, Dennis, "Third Product Line Practice Workshop Report" (CMU/SEI-1999-TR-003, ESC-TR-99-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999.

[Bos00] Bosch, Jan, "Design and Use of Software Architectures: Adopting and evolving a product-line approach", Addison Wesley, May 2000.

[Cle01] Clements, Paul; Donohoe, Patrick; Kang, Kyo; McGregor, John and Northrop, Linda, "Fifth Product Line Practice Workshop Report" (CMU/SEI-2001-TR-027, ESC-TR-2001-027). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001.

[Gre02] Mc Gregor, J. D., Northrop, L. M., Jarrad, S., Pohl, K., "Initiating Software Product Lines", IEEE Software, July / August 2002.

[GS96] Shaw, Mary and Garlan, David, "Software architecture: perspective on an emerging discipline", Prentice Hall, 1996.

[Har01] Harsu, Maarit, "A Survey of Product Line Architectures", Report 23, Institute of Software Systems, Tampere University of Technology, March 2001.

[Jon02] Jones, Lawrence and Soule, Albert, "Software Process Improvement and Product Line Practice: CMMI and the framework for Software Product Line Practice" (CMU/SEI-2002-TN-012). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002.

[Lin02] van der Linden, Frank, "Software Product Families in Europe: The Esaps and Café Projects", IEEE Software, July / August 2002.

[vdH01] van der Hoek, André; Mikic-Rakic, Marija; Roshandel, Roshanak and Medvidovic, Neno "Taming Architectural Evolution", Proceedings of the 6th ESEC and the ACM SIGSOFT (FSE-9), Vienna, Austria, 2001.

[Pus02] Pussinen, Mika, "A Survey on software product-line evolution", Report 32, Institute of Software Systems, Tampere University of Technology, December 2002.

[Sva00] Svahnberg, Mikael and Bosch, Jan, "Issues Concerning Variability in Software Product Lines", in *Proceeding of the Third International Workshop of Software Architectures for Product Families*, Springer Verlag, Berlin, 2000.

[Sva99] Svahnberg, Mikael and Bosch, Jan, "Characterizing Evolution in Product Lines Architectures", in *Proceeding of the Third IASTED International Conference on Software Engineering and Applications (SEA'99)*, Anaheim CA, 1999.

[VdW02] Van der Westhuizen, Christopher and van der Hoek, André, "Understanding and Propagating Architectural Changes", *In Proceedings of the Working IEEE/IFIP Conference on Software Architecture 2002 (WICSA 3)*, Montreal, Canada, August 2002.