Software Architecture Research and Development Project Synopsis

Building product lines from existing software

Department of Computer Science, University of Aarhus

Aabogade 34, 8200 Århus N, Denmark

Group: Bravo

20074842, Lars Kringelbach, lars@kringelbach.com

20064684, Marjus Nielsen, Marjus.nielsen@gmail.com

20074877, Morten Herman Langkjær, morten.herman@gmail.com

20054680, Peter Madsen, pm@chora.dk

<<Date: September 11, 2008>>

Abstract

This document contains the synopsis for an architecture research and development project, which is the final part of the Master in IT course "Software Architecture in Practice". The main focus of the project will be the task of converting existing software into a software product line.

1 Motivation

In many software development organizations, the different products developed share common characteristics. For organizations specialized in a specific field of development the similarities seems to often outnumber the differences and this applies to several types of assets, such as requirements, design and implementation.

A software product line is defined as strategic reuse of all these assets in order to produce a family of products [2]. Developing products from a product line instead of making individual products can have several advantages such as a lower cost, shorter time-to-market, and higher productivity, which improve a company's ability to stay on the market *with lower prices or have greater revenue and lessen the need for hiring more qualified people which are a limited resource. These advantages will however only apply when the product line is successfully implemented, which includes both a software architecture that will fit the company's needs but also an organization that is structured to handle the use of software product lines.*

The project participants all agree that an existing project within their working environment could benefit from being converted to a product line, thus increasing development efficiency and company profits, but have lacked the experience, argumentation and support to pilot such a project within their working organizations. Thus it is the aim for the group to examine the subject of product line conversion in sufficient detail to be able to pilot similar conversions in their respective organizations. It is also the hope that this project can be used as inspiration for people aiming to convert existing software to a product line.

2 Hypothesis/Problem statement

The project will examine theory and practices supporting product lines in order to get hands on experience and to discuss conclusions in current theory.

A lot of software is written for dedicated purposes with no or little support for reuse in other projects, and will often result in source code being copied between projects instead of being reused in a strategic manner.

With focus on the software architecture of a software product line, the scope of this report is to discuss the principles behind, and to show samples of applying architectural patterns and other architectural relevant concepts supporting product lines.

These theories will be applied in practice in order to show how the architecture of an existing software product can be converted into an architecture that supports multiple similar products and hence strategic reuse of source code.

In practice the project will be founded in using an existing software product as a case for trying product line practices. As a case the open source project *DrinkMixer* [1] will be used. Some architectural reconstruction of *DrinkMixer* will be necessary, but the effort will be kept at a minimum and an in-depth discussion on this subject will not be part of the report.

Please note that the practical focus will be on the architectural aspects of software product lines. Other related assets, e.g. requirements, analysis and testing will not be taken into account. No practical work will be done regarding management related assets such as plans and budgets or in assessing an organizations readiness for implementing product lines. The theory section and to some extend the evaluation will cover the theoretical aspects of organizational readiness.

3 Method

In order to examine theory and practices, a presentation of the major theory[1] on product lines will be created. Special focus will be on presenting architectural patterns supporting product line development.

When techniques and patterns have been established, the current architecture of the case project will be identified. As stated above some architectural reconstruction will be needed. This process will focus on building some static views, mainly module- and Component & Connector views, in order to document the architecture 'as implemented'. *DrinkMixer* is limited in size though, so this process is expected to consume only a small part of the project effort.

Once the current architecture is documented in sufficient detail, it will be redesigned to support product development with major reuse of source code in order to fit into a product line.

A software architecture used in a product line need often support behavioral differences between products opposed to only being configurable by parameterization. The analysis of *DrinkMixer* might show that it lacks sufficient business logic to be a well suited candidate for showing this kind of configuration. If this should be the case, some suitable functionality will be added (e.g. a pricing model, printing etc.).

This work will be based on using architectural prototypes, both presenting the theory behind these and to build one supporting a limited number of variability points.

The work needed to build new products using the product line will be assessed in order to comment on the overhead involved in using a product line strategy as opposed to creating individual products. The

[1] Currently [2], [3] and [4] are identified, but an analysis of available literature is expected to extend this list.

comparison will consider the percentage of files and lines that are different between products.

The product line architecture should at least support the same quality attributes as the original product it is being converted from, and have a higher degree of modifiability. Since the original quality attributes are unknown and rediscovering them from the source code is time consuming, we will restrict this portion of the project to a best effort guess.

4 Expected Analyses and Results

Architectural Patterns and styles

Throughout the project various software disciplines that support product lines will be used. We expect to explore some patterns and architectural styles, which can support the transformation to a product line. It is expected that some of the styles and patterns will be tested for suitability by implementing an architectural prototype on the style or pattern. We will concentrate on styles and patterns that may lower coupling in the system, and support product line development. Examples are repository style for decoupling persistence, and MVC for low coupling of the user interface etc.

In order to support several types of products, we may benefit from having a Meta model for the persisted data, in order to interpret the data in different ways, in different user domains.

Prototypes

The prototype is expected to show a substantial amount of reuse across similar products. It will show how different variations can be implemented without violating the architecture which is a necessary property of a software architecture in a software product line.

Discussion of product lines

The report will contain a discussion of product line theory, with a focus on the claims that there is a rather big overhead in product line conversion compared to individual developed products.

Report outline

The following is an outline on the expected sections of the finished report. This is for reference only and is subject to change.

- Motivation

- Hypothesis/Problem statement
- Theory
    - Product line theory
    - Architectural patterns
    - Architectural prototypes
- Reconstruction of *DrinkMixer*
    - *Architectural reconstruction theory brief*
    - *Static views*
    - *Architectural views*
- *Building the product line prototypes*
    - *Adding functionality to DrinkMixer*
    - *Identifying variation points*
    - *Creating strategies*
    - Building architectural prototype(s)
    - Evaluating the prototype
- Evaluation & discussion

References

[1] http://sourceforge.net/projects/drinkmixer/

[2] L. Bass, P. Clements, and R. Kazman. (2003) *Software Architecture in Practice 2nd Edition*. Addison-Wesley, chapter 14 and 15

[3] Clements, P. and Northrop, L. (2002) Software Product Lines.

[4] McGregor, J.D., Northrop, L.M., Jarrad, S., Pohl, K. (2002) Initiating Software Product Lines. IEEE Software 19(4), pp 24-27