Thesis Proposal

Iris Groher
Siemens AG
Otto-Hahn-Ring 6
81739 Munich, Germany

Working Title: Aspect-Oriented Product Line Engineering

Software engineering methods and tools are available and well understood for developing single systems. Most companies however stay within the same domain from project to project or sell products in the same market segment for years. For these companies to be competitive it is essential to reuse artefacts from one product in another in order to improve time to market, achieve systematic reuse goals and improve their product quality. In order to benefit from mass customization it is essential to get control over diverse product configurations.

A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [1]. Basically, product line engineering is about exploiting commonalities among a set of systems while managing the variabilities among them in order to exploit the above mentioned benefits.

Product line engineering typically involves two processes:

Domain Engineering
- Analysis: Scoping and identification of a set of reusable, configurable requirements for the systems in the domain
- Design: Development of a common architecture for the systems in the domain
- Implementation: Implementation of the reusable assets and the instantiation process, e.g. generators

Application Engineering
- Production of concrete systems using the reusable assets developed during Domain Engineering

It is important to note that core assets may not only be components, but also documents, models, project plans, requirements or generators – everything that is common to every product or variable enough to be reused in a number of products that form a family.

Current state-of-the-art in product line development lags behind state-of-the-art in software research with regard to both:

1. each individual development phase in the software life cycle of product families, and
2. the traceability of variations across these development phases and between domain and application engineering.

There is a big gap between research in requirements analysis, architectural modelling and implementation technology, and the industrial practice in software product line engineering. Only few successful software product lines are described in literature [1]. The complexity of a whole family of systems is by far higher than the complexity of a single system. Therefore an approach has to be developed that decreases the complexity by better structuring the problem and solution space.

This thesis aims to exploit aspect-oriented software development (AOSD) [3] in order to overcome the deficiencies in the individual phases and to provide better traceability of variations throughout the life cycle. AOSD techniques address so-called *crosscutting concerns* by providing means for their systematic identification, modularisation, representation, and composition throughout the software lifecycle.

The claim is that variations are often crosscutting as they can have a wide ranging impact across the assets and the product spectrum. This is why AOSD techniques are natural candidates to manage the crosscutting nature of variabilities in a software product line. The approach will provide a holistic treatment of variability in the requirements analysis, design and implementation stages of the software product line lifecycle including traceability of variations. Aspect-oriented requirements engineering techniques provide improved ability to reason about the problem domain while aspect-oriented architecture design techniques help to transform this

reasoning to the solution domain. First class support for abstracting crosscutting concerns at implementation level makes it possible to provide improved alignment with the high level requirements and architecture models of such concerns.

The family members in a product family usually differ in the number and characteristic of the features included in a product. A systematic way to group requirements into features that then form the architectural entities and a seamless tracing of requirements throughout the whole lifecycle is necessary. Features tend to crosscut multiple points in code as well as different levels of the software development lifecycle. AO techniques allow modularizing crosscutting features in design and implementation, which makes tracing of requirements to implementation artefacts a lot easier.

In this thesis an aspect-oriented product line framework will be developed for identifying, specifying, and managing variability and features from requirements analysis, architectural design to implementation. For this purpose pre-existing or currently elaborated work in the fields of AO requirements engineering, AO design and implementation languages will be evaluated and incorporated into the framework where appropriate. There exist already a number of aspect-oriented requirements engineering, design, and implementation approaches [4] [5]. Though their applicability for software product line engineering has not yet been evaluated. In this thesis unsolved problems in existing approaches will be identified and possible solutions presented.

The thesis is divided in different work packages:

- Existing **Aspect-oriented requirements engineering** techniques are studied and evaluated for their applicability in domain analysis. It is investigated how to capture commonalities and variations making use of aspect-oriented concepts and techniques. A systematic approach shall be provided that helps domain analysts in grouping of requirements into features. The goal is to (semi) automatically develop feature models out of requirements specifications. It is also investigated how aspect-orientation can improve feature modelling and how crosscutting features can be modelled at the requirements level.

- Existing **Aspect-oriented architectural modelling and design** techniques are studied and evaluated for their applicability in domain design and especially for their ability to manage variability. The evaluation will specifically focus on accounting for variations and commonalities in the assets of a product line.

- Existing **Aspect-oriented implementation** techniques are studied and evaluated for their capabilities in mastering modularization of (crosscutting) features and their support for capturing variability. The evaluation will be structured around selected features of an implementation technology, such as support for expressing reusable assets in a modular way, support for variability or composition.

- A **traceability** framework is provided that helps in making the relationship between requirements, the derived architecture, design and implementation artefacts explicit. It provides forward and backward traceability from requirements to implementation. It should be capable of representing explicitly decisions taken throughout the above mentioned phases of the software product line lifecycle.

The approach will be presented using a large industrial case study and some possible candidate studies have already been investigated (e.g. a system family of toll systems). The case study will be used to analyze the challenges, elaborate the tools and technologies to reach the goals of the thesis and to evaluate the results.

Close collaboration with Siemens AG gives the opportunity to participate in EC funded projects on AOSD and product lines to leverage pre-existing work within academic partners throughout Europe.

[1] P. Clements and L. Northrop. *Software Product Lines: Practices and Patterns.* Addison-Wesley, 2002.
[2] P. Zave, “FAQ Sheet on Feature Interaction”, http://www.research.att.com/~pamela/faq.html, 2005.
[3] T. Elrad, S. Clarke, and M. Aksit. *Aspect-Oriented Software Development.* Addison-Wesley, 2004.
[4] AOSD-Europe. Survey of Aspect-Oriented Analysis and Design Approaches. http://www.aosd-europe.net/documents/index.htm
[5] AOSD-Europe. Survey of Aspect-Oriented Languages and Execution Models. http://www.aosd-europe.net/documents/index.htm

List of publications and activities

Workshops

- Co-organiser of the *1st Workshop on Models and Aspects* (held in conjunction with ECOOP 2005)

Posters

- *Advantages of and Challenges for AOSD for Product Line Development* at the Student Research Extravaganza at the AOSD conference 2004

- *Improved Product Line Development with Generative Approaches* at the European Interactive Workshop on Aspects in Software (EIWAS) 2004

Papers

- Iris Groher and Stefan Schulze. *Generating Aspect Code from UML Models.* In Proceedings of the Workshop on Aspect-Oriented Modeling. AOSD conference 2003.

- Iris Groher and Thomas Baumgarth. *Aspect-Orientation from Design to Code.* In Proceedings of the Workshop on Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design, AOSD conference 2004.

- Iris Groher. *Managing Variabilities with Generative Approaches*. In Proceedings of the Workshop on Managing Variabilities Consistently in Design and Code, OOPSLA 2004.

- Iris Groher, Stephan Bleicher and Christa Schwanninger. *Model-Driven Development for Pluggable Collaborations.* (To appear) in Proceedings of the Workshop on Aspect-Oriented Modeling, MoDELS 2005.

- Iris Groher, Stephan Bleicher and Christa Schwanninger. *Designing Features as Pluggable Collaborations.* (To appear) in Proceedings of the Workshop on Aspects and Product Lines, SPLC-Europe 2005.

- Iris Groher, Vaidas Gasiunas, Christa Schwanninger and Klaus Ostermann . *Teile und Herrsche: Aspektorientierte Softwareentwicklung mit CaesarJ*. To appear in Java Spektrum 2006.