Controlling the variant explosion

- Enforcing stability in highly configurable large scale software

Master thesis popular description

Anders Hellström & Björn Pileryd, 2005-09-28

The mobile phone industry has changed rapidly during the past few years. The relation between phone producers, operators and customers has been a “take it or leave it” situation where the producers would offer phones and the operators would decide if to adopt the product or not. The number of mobile phone users has increased tremendously and the market is intensified. The product lifetime of a mobile phone is now down to about a year in Europe and just a few months in Japan. The tightened competition and increased availability of mobile phones has shifted the power to shape the product more towards the operators. Now operators want to offer services that other operators do not have. In order to be successful on the mobile phone market Sony Ericsson Mobile Communications, SEMC, must be able to quickly meet the demands and rapidly produce new products with new features.

To meet the different demands from the operators and customers the software in the phone needs to be highly configurable. The difference between the demanded functionalities is often relatively small compared to the total amount of source code in the software. To reduce the time and resources needed to produce these different software variants, existing code is reused.

The ideal situation for SEMC when setting up a new mobile phone variant would be to have a shelf of available feature components to choose from to compose the desired functionality. These feature components would all be compatible and no matter how they were combined, they would always produce perfectly working phone software. The real situation is however different. Because today’s phones offer many applications and services besides phone calls the functionality has become very complex. Features are not totally modular as in the ideal situation. A feature often interacts with other features and expects the existence and a certain behavior of these. The practical symptoms of this are that when a new variant of a product is created the software does not always build, and even if it builds, it does not always work.

SEMC now faces the problem of keeping the software components as modular as possible and in the same time allowing functionality to interact closely. On top of this the software must be configurable on a detailed level. To achieve this the software development environment at SEMC uses configuration variables specified in configuration files to set the desired functionality of the software.

This master thesis work aims to further strengthen the software development environment at SEMC. The work is divided into three parts. The first objective is to investigate methods to configure software variants of the phone in a more dynamic way than it is done today and develop a tool to compose and build software variants. The second is to investigate methods to gather and visualize dependencies between configuration variables, and the third objective is to investigate how the dependency information can be used, e.g. to form a framework of rules to validate software configurations.

The result of the first objective is a program called Dynamic Variant Tool which uses current functionality in the development environment to make it easier to create, build and rebuild variants of existing products.

During the analysis of the second objective, how to visualize dependencies, it became apparent that the dependencies between configuration variables can be visualized in different ways. Two dependency types were defined and methods to visualize these are presented in the thesis report. These methods are suitable for different situations, one of them gives a general overview and the other presents a detailed description of the configuration variables and their dependencies.

Focus on the third objective of the thesis work was shifted to investigate whether a framework of rules can be used to analyze the configuration files. The framework shall consist of separate rules that can be used to validate and verify the configuration files in different ways. We suggest a number of rules along with issues and benefits of these. A categorization of the rules and how to implement the framework is also presented in the thesis report.

The overall results of the thesis work is a number of suggestions and recommendations on how to enforce stability in the highly configurable large scale software at SEMC. The risks and costs of implementing these recommendations are still to be carefully and thoroughly investigated, but we are certain that major benefits can be achieved.