# Studies on Architecture Refactoring for Software Product-Line Development

Takashi Maki

School of Information Science,
Japan Advanced Institute of Science and Technology

December, 2013

## Abstract

In recent years, it has become increasingly important to maintain architecture in product-line development (PLD), mainly because of the rapid changes in market requirements and technical environments. In PLD, architecture maintenance is a more complicated and difficult process compared to conventional software development because architecture is key to achieve large-scale reuse in developing a product family. In architecture maintenance, we have to consider both the reference and implemented architectures. Here, reference architecture is a design intention that constraints the implementation, and implemented architecture is an abstract expression of the existing implementation.

Architecture maintenance includes both keeping the conformance of implemented software architecture with the reference architecture and changing the reference architecture to meet new requirements. These architecture changes are modifications of software structure without changing the major feature of the product family. Thus, we call such modifications architecture refactoring.

In PLD, requirements for reference architecture can change during the development of the product family because the development period lasts longer than that in non-PLD. Moreover, the implemented architecture can deteriorate over the development of multiple products. Therefore, we can organize architecture refactoring more efficiently by separately considering the implemented and reference architectures refactorings. In this study, we propose a decision taking method for architecture refactoring that considers both the implemented and reference architectures separately.

The main characteristic of this method is utilizing the portfolio analysis of the problem factor to organize the architecture maintenance strategy. Furthermore, we verified the effectiveness of the proposed method by applying actual project data to the proposed method retroactively.

**Key Words:** software architecture, architecture refactoring, product-line development, decision making, portfolio analysis