

Title	プロダクトライン開発におけるアーキテクチャリファクタリングの研究
Author(s)	牧, 隆史
Citation	
Issue Date	2013-12
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/11933
Rights	
Description	Supervisor:Defago Xavier, 情報科学研究科, 博士

氏 名	牧 隆史
学 位 の 種 類	博士(情報科学)
学 位 記 番 号	博情第 290 号
学 位 授 与 年 月 日	平成 25 年 12 月 20 日
論 文 題 目	Studies on Architecture Refactoring for Software Product-Line Development (プロダクトライン開発におけるアーキテクチャリファクタリングの研究)
論 文 審 査 委 員	主査 小川 瑞史 北陸先端科学技術大学院大学 教授 落水 浩一郎 同 教授 青木 利晃 同 准教授 岸 知二 早稲田大学 教授 深澤 良彰 同 教授

論文の内容の要旨

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.

論文審査の結果の要旨

近年類似した製品系列のソフトウェアを全体最適の視点から体系的な再利用を行って開発するプロダクトライン開発(PLD)が産業界でも広まりつつある。こうした大規模な再利用においては、共通の(ソフトウェア)アーキテクチャを設定することが効果的な再利用資産の開発や活用のために必須となるが、時間の経過とともに様々な理由でアーキテクチャが崩れていく現象が避けられず、問題となっていた。こうした崩れたアーキテクチャを改善する作業は製品群全体に影響を及ぼしコストが高くつくため、限られたリソースの中で徹底的な改善を行うことは困難である。したがって現実には効果的と考えられる改善作業を選択的に実施するが、従来その判断は経験者の勘と経験に基づいたアドホックなものとなりがちであった。

本研究はそうした改善作業をアーキテクチャのリファクタリングと捉え、効果的なリファクタリング項目（改善項目）を選択するための意思決定手法を提案するものである。ここでは本来意図されたあるべき姿としての参照アーキテクチャと、実際のソフトウェアが具現化しているあるものの姿としての実装アーキテクチャの二つに注目し、アーキテクチャに起因するプロジェクト運営上の問題をこの二つのアーキテクチャの問題として整理し、その性格をポートフォリオ分析することでリファクタリング項目を優先度付けするという新規な技法を活用している。また本手法は実務者がプロジェクトにおいて日常的に捉えやすいQCD上の気づきをアーキテクチャリファクタリングのきっかけ（匂い）とし、通常のプロジェクトにおいて獲得しやすい一般的なメトリクスに基づいて分析するなど、実務的で分かり易く、現場で利用しやすい手法として整理・体系化されている。さらに過去に現場で行われたデジタルカメラのプロダクトライン開発の実データを活用し、手法を再適用することによって、手法に基づくリファクタリング項目の選択判断や、そこで使われているポートフォリオ分析の妥当性を確認している。

以上、本論文は、プロジェクト上の現象からアーキテクチャ上の課題へとブレイクダウンし、それをポートフォリオ分析することでリファクタリング項目の優先度付を行うための新規かつ実用性のある手法を提案するもので、学術的に貢献するところが大きい。よって博士（情報科学）の学位論文として十分価値あるものと認めた。