

論文 / 著書情報
Article / Book Information

題目(和文)	ソフトウェアプロダクトラインにおけるフィーチャの相互作用の実装に関する研究
Title(English)	A Study on Implementations of Feature Interaction in Software Product Lines
著者(和文)	武山文信
Author(English)	Fuminobu Takeyama
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第9359号, 授与年月日:2013年12月31日, 学位の種別:課程博士, 審査員:渡辺 治,千葉 滋,増原 英彦,首藤 一幸,脇田 建
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第9359号, Conferred date:2013/12/31, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	審査の要旨
Type(English)	Exam Summary

論文審査の要旨及び審査員

報告番号	甲 第 号		学位申請者		武山 文信	
		氏 名	職 名		氏 名	職 名
論文審査員	主査	渡辺 治	教授	審査員	首藤 一幸	准教授
	審査員	千葉 滋	連携教授			
		増原 英彦	教授			
		脇田 建	准教授			

論文審査の要旨 (2000 字程度)

本論文は「A Study on Implementations of Feature Interaction in Software Product Lines」(ソフトウェアプロダクトラインにおけるフィーチャの相互作用の実装に関する研究)と題し、英文5章から構成されている。

本論文では、ソフトウェアプロダクトラインの開発手法の1つであるフィーチャ指向開発について取り組み、フィーチャの間に生じる相互作用をプログラムで表現するための言語機構とそれを備えたプログラミング言語を提案している。FeatureGluonJ と呼ばれる言語では、関連するフィーチャを実装するための継承機構を導入し、この継承機構を用いて、関連するフィーチャ間の相互作用を少ないコード量で実装できるようにしている。2つめの言語 Airia では、フィーチャの実装にアスペクト指向プログラミング (AOP) を用いたときに生じる相互作用であるアスペクトの衝突を解消するための言語機構 resolver を提供する。

第1章「Introduction」では、研究の背景と動機、そして手法の全容を紹介している。ソフトウェアプロダクトラインの開発には、フィーチャと呼ばれる、エンドユーザ視点のソフトウェア製品の性質に注目することが有用である。またプログラムをフィーチャ毎のモジュールに分割することで、ある製品に必要なフィーチャを選び、対応するモジュールをコンパイルすると、最終的な製品を生成することができる。フィーチャ毎にモジュールを分割する手法には、フィーチャ指向プログラミング (FOP) 言語と AOP 言語がある。

第2章「Feature-oriented Software Development」では、フィーチャを中心としたソフトウェア開発の手法を紹介し、FOP と AOP でプロダクトラインを開発する際に生じるそれぞれの相互作用の問題を述べている。相互作用とは、プロダクトラインを構成するために組み合わせられたフィーチャにまたがる動作のことである。このようなフィーチャの組み合わせが FOP で正しく動作するためには、プログラマが注意深く相互作用を実装しなければならない。相互作用の実装は、その組み合わせの両方が同時に使用されるときにのみ、組み込まなければならないため、デリバティブと呼ばれるモジュールに独立して実装されるべきとされてきた。しかしながら、この手法には問題があり、プログラマが事前にすべてのフィーチャの組合せに対して、多量のデリバティブを事前に用意しておかなければならない。一方、AOP をプロダクトライン開発に用いる場合も、クラスの外側から既存のメソッドの定義を変更するアドバイスと呼ばれる機構の衝突が問題となる。この衝突はそれぞれのフィーチャを実装する複数のアドバイスが同時に同じメソッドを変更しようとするときに発生する。従来の衝突の解消方法は、アドバイスに優先度を明示的に与えて、順に実行することであった。しかしながら、アドバイスの組合せによっては適切な順序が存在せず、それぞれのアドバイスを順に実行できるように修正しなければいけない場合があり、従来の方法ではうまく衝突を解消できない。

第3章「FeatureGluonJ」では、新しいフィーチャ指向プログラミング言語である FeatureGluonJ を提案している。フィーチャが汎化・特化関係を持つことに着目し、継承を用いて関連するフィーチャを実装できるようにする。またこのとき上位のフィーチャを下位のフィーチャのインタフェースとして用いることで、下位のフィーチャ群と他のフィーチャの相互作用を汎用的に記述できるようになる。これにより FOP において、その相互作用を記述するために多数のデリバティブを実装しなければならない事態を回避できる。

第4章「Airia」では、アドバイスの衝突を解消するための新しい種類のアドバイス resolver を持つ AOP 言語 Airia を提案している。この resolver は AOP におけるアドバイスの衝突を解消するために、衝突しているアドバイスの一部を選択して実行し、その実行結果を合成する機能を提供する。また resolver 自体もアドバイスであるため、resolver と他のアドバイスとの衝突も、同様の方法で解消できる。さらに本章では、従来の研究で用いられている例の多くを resolver でも記述できることを示している。

第5章「Conclusion Remarks」では、本論文のまとめを述べている。

以上のように本論文は、フィーチャ指向によるプロダクトライン開発の有用性を高める新しい言語機構を提案し、評価によって有効性を示している。これらの成果は、プログラムのモジュール化によるソフトウェアの品質向上に寄与し、理学的貢献するところが大きい。よって本論文は、博士(理学)の学位論文として、十分な価値があるものと認める。