

Title	フィーチャー指向ソフトウェアにおける増分的一貫性 検証
Author(s)	Nguyen, Truong Thang
Citation	
Issue Date	2005-09
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/984">http://hdl.handle.net/10119/984</a>
Rights	
Description	Supervisor:片山 卓也, 情報科学研究科, 博士

## 論文概要

オブジェクト指向ソフトウェア技術はソフトウェア開発の非常に有力なアプローチである。オブジェクト指向技術はソフトウェア産業の発展に大きく貢献しているが、まだいくつかの欠点が存在し、特に良く知られている問題として、複数のオブジェクトに横断的に存在する関心事[24]を捕らえにくいことが挙げられる。横断的な要素は、これらオブジェクト間の関心事を複雑なコードで実装していることが原因である。結果として、コストが非常に高くなる。

これらの問題を解決するために、オブジェクト指向技術の拡張として、アスペクト指向ソフトウェア開発(AOSD)が提案されている。AOSDは高い再利用性やその他多くの優れた特徴を持ち、これからのソフトウェア開発方法論として有望である。本論文では、関心事[34]、アスペクト[14]、フィーチャー[25, 26] コラボレーション[8, 24, 32]、ハイパースライス[33]およびコンポーネントなどの用語をそれぞれ互換性のある概念として扱う。これらの用語は、ソフトウェアのライフサイクルの様々な段階における、関心事の粒度のレベルの違いに目を向けたものであり、等価なものとして扱うことができる。より具体的には、本論文は、複数のオブジェクトのコラボレーションによってシステムが提供するサービスや機能であるハイパースライスの標準的な定義を与える。

フィーチャー指向ソフトウェアはAOSDの特別な場合であり、フィーチャーはシステムの機能面の要求に対応する。本質的には、システムはいくつかのフィーチャーの複合として構成される。各フィーチャーは独立して仕様記述、実装、保守が行われる。これらのフィーチャーは相互に、あたえられているインターフェースを経由して通信する。この設計スタイルを基礎とすれば、システムが特定のフィーチャーを持つか持たないかというものの決定に従って、システムを柔軟に構成することができる。

しかしながら、このような概念的ソフトウェア開発方法論を実現するために、独立的に仕様記述と実現をするこれらフィーチャーがそれぞれを組み合わせたとき、それらに不整合が起こらないことの保証は重要な課題である。これをフィーチャーの一貫性の問題と呼ぶ。本論文は、この問題および関連する諸問題を形式的手法で取り扱う。本論文は大きく以下の3つの部分から成る。

最初に、一貫性の問題の形式化を行う。各フィーチャーを状態遷移機械としてカプセル化した、フィーチャー指向ソフトウェアの形式的モデルを導入する。このフィーチャーの形式的モデルは、他のフィーチャーを接続するためのインターフェースの表現も提供する。いくつかのルールによって、異なるフィーチャーのインターフェース状態がそれぞれマッピングされる。この

ことによりフィーチャーを合成して対象となるシステムを構成することが可能となる。

2 番目は、本論文の中核であるフィーチャー間の一貫性検証の理論的基礎付けである。あるフィーチャーがその実行中に、もう一方のフィーチャーが持つ CTL 性質に違反しないとき、そのフィーチャーは他方と**一貫性がある**という。検証の手法-Open Incremental Model Checking (OIMC) と呼ぶ-はモデル検査手法[5, 17]に基づいて行われる。OIMC とその他のモジュールを考慮したモデル検査手法[10, 15, 17, 28]との違いは、増分的オープン性(Incremental Openness)である。本論文の手法では、システムはつねに進化し続けると考える。典型的な進化のシナリオは新しいモジュールを既存のシステムに一貫性を保ちつつ追加することである。このような状況では、新しい手法では拡張部分のモジュール側でのみモデル検査を実行すればよい。一方、伝統的なモデル検査の手法では、システムがモジュール化されていたとしても、システム全体を検査し直さなくてはならない。モデルチェッキングの分野において、本論文の手法は非常に良い性質を持っているといえる。本論文では OIMC の健全な理論的基礎を提案する。この目標は、拡張側のコンポーネントが既存のコンポーネントの特性に違反しないような必要条件を明らかにすることである。さらに、増分的な検証アプローチのスケーラビリティと健全性について議論する。フィーチャー合成の非常に特別な場合を除いて、OIMC によって得られる結果は健全である。また、整合性条件を仮定した場合、どんな拡張の発生についても OIMC の計算量は変化しない。すなわち、スケーラビリティが高いといえる。

本論文の最後の部分は、以前の理論的基礎に基づくいくつかの応用、コンポーネントの仕様と合成など、を要約する。さらに、フィーチャー指向ソフトウェアの形式的仕様モデルとその実装との関係を示す。より詳細には、形式的モデルを具体的なプログラミング言語(JAVA)に対応付ける、基本的なコード変換メカニズムをあたえる。フィーチャーが独立して仕様記述されているので、対応するコードは変換ルールによってそれぞれ独立して生成することができる。最後に、各フィーチャー対応するコードに対し、合成[33]やウィーブ[35]を行って一つにまとめる。