

非機能要件が与えるソフトウェアの複雑さへの影響評価のためのフィーチャモデルと CDG の利用

邵 楠[†] 中里 友昭[‡] 森谷 大輔[†] 渡辺 晴美^{†‡}

[†]東海大学 情報通信学研究科 [‡]東海大学 情報通信学部

Abstract: Measurement of the complexity of the model plays important roles of increasing the maintainability. Especially, measuring impacts of non-functional requirements is significant, many of such requirements become crosscutting concerns. To measure the complexity of non-functional requirements, we will introduce an approach of combining CDG with the feature model. For evaluating the approach, we will apply it to a development of automatic vacuum cleaner. The evaluation will clarify the impact of non-functional requirements on the complexity of models.

1. はじめに

モデルの複雑さの計測は、保守性を高める上で重要である。モデルの複雑さが増す原因として横断的関心事がある。横断的関心事は、その名のとおりクラスに横断的であることから、関連数増加、すなわちモデルの複雑さが増すことになる。横断的関心事の多くは、非機能要件に起因していると言われている。

非機能要件を扱うモデリング技術にフィーチャモデルがある。フィーチャモデルとは、フィーチャ間の関係を木構造で表したグラフであり、ソフトウェアプロダクトライン開発のために提案されたグラフである[1]。フィーチャモデル特徴は、機能フィーチャと非機能フィーチャ（非機能要件）を分析できる[2]。

一方、クラスの複雑さを計測する方法として CDG (Class Dependency Graph) がある。CDG とは、クラス間の依存関係を表したグラフである。グラフ中にあるノードはクラスに対応し、矢印は関連に対応する。クラス間の関連として継承を重視している。オーバーライトやポリモフィズム等、振る舞い予測を難しくする要素を含むためである[3]。CDG はマイクロソフト.net や google で利用されているほど著名な方法である。

フィーチャモデルと CDG の組み合わせることで非機能要件がモデルへ与える複雑さを測ることができる。我々は、横断的関心事を扱うアス

ペクト指向やコンテキスト指向に関する研究を行ってきた。アスペクトやコンテキストの抽出に際し、非機能要件が与えるモデルへの影響を調べることは重要なことである。

本稿では、非機能要件がモデルへ与える影響を調査するために、フィーチャモデルと CDG を関連付けたツールを紹介する。以下の第2章では、フィーチャモデルと CDG グラフを描く手順を紹介する。第3章では、掃除ロボットシステムを対象として実験が行ったことを記す。第4章では、本稿のまとめと今後について述べる。

2. フィーチャモデルに基づく CDG 拡張グラフ

フィーチャモデルに基づく CDG 拡張グラフ構築の手順について、以下に述べる。

- (1) フィーチャの抽出：フィーチャモデルを用いて機能フィーチャと非機能要件を抽出する。
- (2) フィーチャモデルの構築：定義できたフィーチャによってフィーチャモデルグラフを描く。
- (3) クラスの抽出：クラス図を構築するために、クラス、属性、メソッドの候補をフィーチャから抽出する。
- (4) クラスの構築：手順(3)で抽出したクラス、属性、メソッドを分類整理したクラス図を構築する。
- (5) クラス図から CDG を構築する：手順(4)で作成したクラス図をもとに、クラス間の依存関係を CDG によって表す。
- (6) 複雑さを測る：CDG グラフに基づいてクラス間の関係を計測し、プログラム複雑さを測って品質を評価する。CDG の式を使って複雑度を計測する。

3. 実験—掃除ロボットシステム

第2章拡張グラフの有効性を確認するために、

An Impact Analysis for Non-functionality Based on Feature Model and Class Dependency Graph

SHAO NAN[†] NAGAZATO TOMOAKI[‡] MORIYA DAIZUKE[†] WATANA BEHARUMI^{‡†}

[†]Information and Telecommunication Engineering Graduate School of Tokai University

[‡]Department of Information and Communication Technology of Tokai University

実験を行った。実験対象はコンテストで使った掃除ロボットシステムである。主なシステムの機能は、掃除ロボットが障害物のある環境で走って、障害物の位置と大きさを把握して、充電ステーションに帰還することである。以下に本実験の結果を述べる。

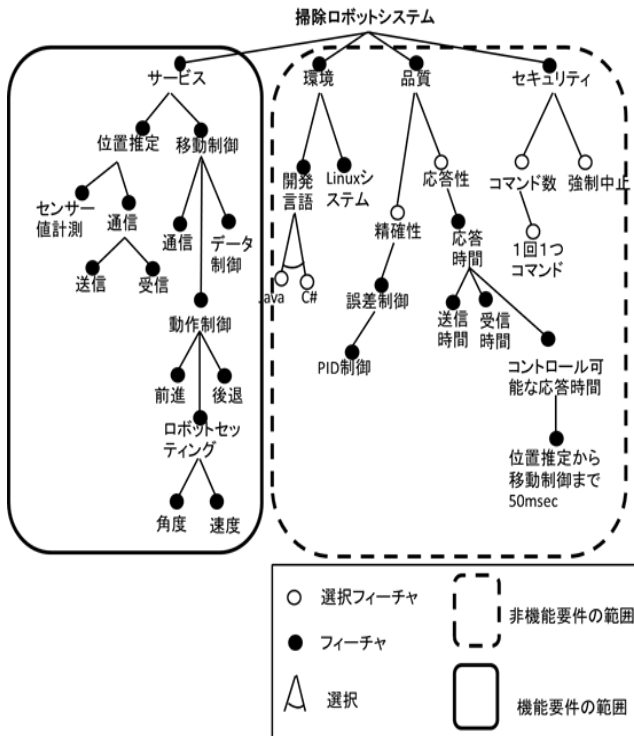


図1 フィーチャモデルグラフ

図1は、第2章に手順(1)と(2)の通りに、フィーチャモデルを使って、掃除ロボットシステム機能要件と非機能要件を出来る限り分析した。掃除ロボットシステムを対象として、構築したフィーチャモデルグラフである。掃除ロボットシステムの機能フィーチャは図1の実線の四角形の部分で示している。点線の四角形部分は、非機能要件である。非機能要件は環境、品質、セキュリティである。

図2は、第2章の手順(3)と(4)の通りに、機能要件と非機能要件(フィーチャ)に基づくクラス図を構築した。フィーチャモデルグラフとフィーチャに応じるクラスのグラフである。一点鎖線の矢印と一点鎖線の四角形は、非機能要件に起因として行った横断的関心事である。クラスに横断的な関連があることで、複雑さが増えている。構築したクラスに基づいて、クララ図からCDGグラフに変更した。

最後に手順(6)に従って、複雑度の式を使って掃除ロボットシステムの複雑さを計測した。計算した結果によって、CI(Sensor, Command)に対して複雑度は28という結果を得た。実験の結果によって、非機能要件がモデルの複雑さに

影響することがわかった。掃除ロボットシステムの複雑度は28となった原因は、非機能要件が存在している、各クラスの横断関心事が多くなって、関連の数が増えることである。

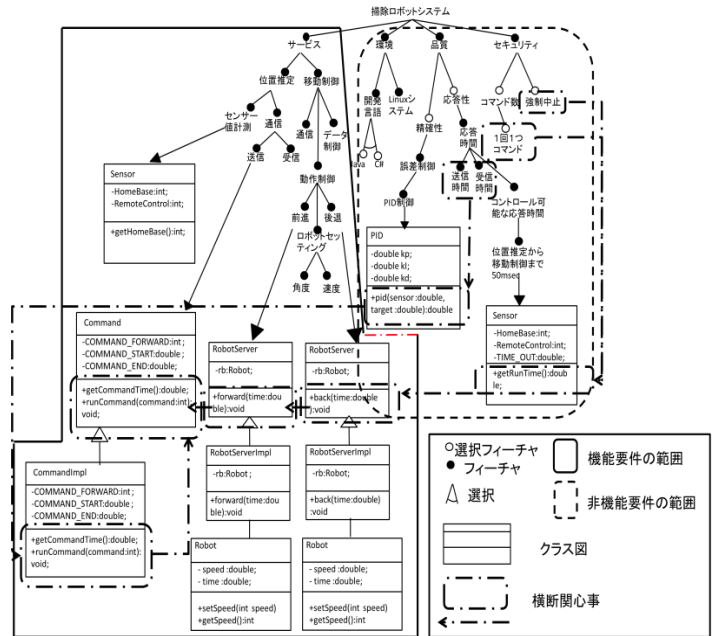


図2 フィーチャモデルグラフとクラス図

4. おわりに

本稿では、実験の内容と結果を示し、実験の結果によって、非機能要件が与えるモデルへの影響を明らかにした。我々が取り組んでいるアスペクトやコンテキストのレイヤ抽出に、本研究の成果を適用する見通しがたった。

今後は、非機能要件がシステムへ与える複雑さについて、実践的なシステムに適用し評価する予定である。

参考文献

[1] Kyo C. Kang, Jaejoon Lee, "Feature-oriented product line engineering" IEEE Computer Society, pp.58-65, (2002)
 [2] Kyo C.Kang, "A Feature-Oriented Approach to Developing Dynamically Reconfigurable Products in Product Line Engineering" 2006 10th International Software Product Line Conference, pp.10-140, (2006)
 [3] Benoit Baudry, Yves Le Traon, "Testability Analysis of a UML Class Diagram" 2002 Proceedings of the Eighth IEEE Symposium on Software Metrics, pp.54-63, (2002)