

アスペクト指向ソフトウェアアーキテクチャに基づく PLSE に関する研究

－ プリンタ制御ソフトウェアを例にして －

2004MT100 鈴木 真貴 2004MT113 浦野 彰彦

指導教員 沢田 篤史

1 はじめに

近年、組み込みソフトウェア開発において、ユーザ要求の多様化に伴い、製品の多機能化が進み、ソフトウェア構造が複雑化している。そのため、開発時間、コストが増加しており、再利用性を考慮したソフトウェア開発が求められている。

再利用性を考慮した系統的なソフトウェア開発をおこなう手法として、Product Line Software Engineering(以下、PLSE)が提案されている。PLSEにおけるドメイン分析を支援する開発方法論である Feature Oriented Product Line Engineering(以下、FOPLE)では仕様モデルとしてフィーチャ図が提案されている。PLSEの問題点として、ドメイン工学における要求仕様とアーキテクチャの関連が明確でないことが挙げられる。また、既存の仕様モデルではフィーチャの定義が不明確であること、記述する非機能フィーチャの抽象度が高く、ほかのフィーチャとの関連を読み取ることが困難であることが挙げられる。

一方で、本研究室では組み込みソフトウェアのアスペクト指向ソフトウェアアーキテクチャスタイル(以下、E-AoSAS++)が提案されている。E-AoSAS++を用いることで、再利用性を考慮した適切なモジュール化が可能となる。

本研究は、PLSEのドメイン工学における問題点を解決することを目的とする。既存の仕様モデルにおける問題点を挙げ、再定義をおこなう。例題として、プリンタ制御ソフトウェアの分析をおこない、再定義した仕様モデルを用いて分析結果の整理をおこなう。プリンタ制御ソフトウェアを事例としたフィーチャ図とアーキテクチャの対応関係を示し、対応関係の一般性に関する考察をおこなう。

2 研究の背景

2.1 PLSE

PLSEは、再利用部品を開発、管理することで、ソフトウェアの系統的な開発を支援する開発方法論である。PLSEにおける3つの活動を以下に示す。

- ドメイン工学
ユーザ要求を基に、コア資産の開発設計をおこなう
- アプリケーション工学
ドメインエンジニアリングで開発したコア資産を基に、ソフトウェア製品の開発をおこなう

- 管理

コア資産および製品の保守、プロジェクト管理をおこなう

2.2 E-AoSAS++

本研究では、PLSEにおけるアーキテクチャ作成に適用するアーキテクチャスタイルとして、E-AoSAS++を用いる。E-AoSAS++では、組み込みソフトウェアを並行状態遷移機械(以下、CSTM)の集合であると定義している。複数の並行状態遷移機械が、互いにメッセージを送る事で協調動作する。

2.2.1 UMP

E-AoSAS++におけるモジュール化の手法として、Universal Modularization Pattern(以下、UMP)が提案されている。UMPでは、CSTMの協調動作を、ポリシーによる切替えで表現する。ポリシーには、コンポーネントをどのように取り扱うか記述することができる。図1にUMPの記述例を示す。

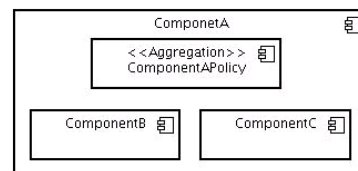


図1 UMP記述の例

2.2.2 E-AoSAS++に基づくアーキテクチャ

E-AoSAS++に基づくアーキテクチャとして以下を示す。

- コンセプチュアルアーキテクチャ
シナリオ及びユーザ要求から、自然に認識できるハードウェア部品間の関連を示す。
- プロダクトラインアーキテクチャ
プロダクトライン全体における、ソフトウェアの構造を示す。
- プロダクトアーキテクチャ
フィーチャ図上で機能の選択がおこなわれた結果、プロダクトラインアーキテクチャから必要なコンポーネントが選択されたもの。プロダクト毎に作成される。

2.3 FOPLE

PLSEにおけるドメイン分析を支援する目的で提案されているのがFOPLEである。FOPLEではユーザ要求の中で、ソフトウェアの開発者が実現可能なものをフィーチャとして定義し、抽出したフィーチャをフィーチャ図

を用いて整理することを提案している。

フィーチャ図を用いて、要求から抽出したフィーチャを、図2に見られるような関係と4つの階層に分けることで、要求仕様のモデル化をおこなう。

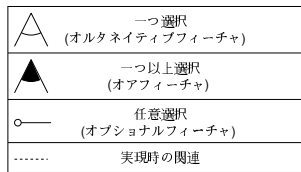


図2 フィーチャの関係

フィーチャ図における4つの階層と各層におけるフィーチャの定義は次に示す通りである。

- 特性層
機能特性、非機能特性に関するフィーチャを記述
- 操作環境層
ハードウェアに関するフィーチャを記述
- ドメイン技術層
開発ドメインに特化した技術を記述
- 実現技術層
開発環境，一般的な規格等の技術を記述

3 既存仕様モデルにおける問題点と解決策

3.1 既存仕様モデルにおける問題点

既存の仕様モデルにおける問題点は次の通りである。

- 非機能フィーチャの記述
FOPLEでは、非機能フィーチャが使用性や信頼性等、抽象度の高い状態で記述されており、ほかのフィーチャとの関連をフィーチャ図上で認識することが困難である。
- 特性層におけるフィーチャの定義
FOPLEでは、特性層において抽出するフィーチャを、機能特性と非機能特性に関連するものと定義している。しかし、この定義ではフィーチャとして抽出するものが不明確である。

3.2 非機能フィーチャの詳細化

既存フィーチャ図における非機能フィーチャの問題点は、フィーチャ図上で関連を読み取れる抽象度になるまで詳細化することで、解決できると考えられる。

3.3 フィーチャ定義の明確化

フィーチャの定義が不明確であるという問題点に対し、フィーチャ図の特性層を、特性層とアトミック特性層の二つの階層に分けることで、上記の問題点を解決する。新たに定義した二つの階層におけるフィーチャの定義を次に示す。

- 特性層
ユースケース図を作成し、ユースケース図に記述されたユースケース、それに関連する抽象度の高い非機能をフィーチャとして抽出する。
- アトミック特性層

特性層に現れたフィーチャを実現する際に、システムが必要とする機能、詳細化した非機能をフィーチャとして抽出する。

上記のように再定義することで、ドメイン分析をおこなう過程において、ユースケースと特性層のフィーチャを関連づけ、フィーチャの定義を明確化できる。

4 プリンタ制御ソフトウェアの分析

プリンタ制御ソフトウェアの分析を、機能、非機能と構成部品に対しておこなった。本研究では印刷、コピー、スキャンなどの機能を持つ複合プリンタを事例とする。分析結果をフィーチャ図を用いて整理する。

4.1 機能の分析

機能の分析を目的として、プリンタ制御ソフトウェアのユースケース図を作成する。作成したユースケース図から認識できるプリンタの機能は次の通りである。

- 印刷、スキャン、コピー
- プリンタの状態、エラーの表示
- 消耗品管理 (インク、用紙)

これらの機能を詳細化したものは次の通りである。

- 印字、送紙
- 印刷データ受信、データ読取り
- 出力、出力データ受信
- 残量監視 (インク、用紙)、エラー送信

4.2 非機能の分析

機能分析の結果に基づいて、プリンタ制御ソフトウェアに要求される非機能について分析をおこなう。考えられる非機能を次に示す。

- 信頼性
- 使用性
- 実時間性

本研究では、実時間性、信頼性について詳細化をおこなった。詳細化をおこなうことで、次の特性が抽出できる。

- 印刷速度
- 送紙速度、印字速度
- 印刷品質維持
- インクヘッドクリーニング

4.3 構成部品の分析

構成部品の分析結果を次に示す。

- インクカートリッジ、ノズル
- プラテン (送紙ローラー)
- 印刷データ受信装置
- スキャナ
- ディスプレイ
- 用紙、インク残量センサ
- USBポート、LANポート
- インクヘッド監視装置、ヘッドクリーニング装置

4.4 フィーチャ図の作成

これまでに分析した結果を、我々が提案するフィーチャ図の定義に基づいて整理したものを図3に示す。

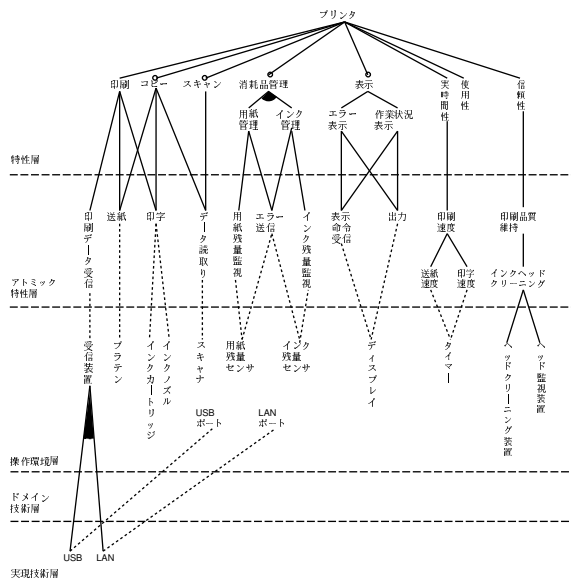


図3 プリンタ制御ソフトウェアのフィーチャ図

5 フィーチャ図とプロダクトラインアーキテクチャの対応関係

四章で分析したプリンタ制御ソフトウェアを事例としてフィーチャ図とプロダクトラインアーキテクチャの対応関係について考察する。プロダクトラインアーキテクチャはE-AoSAS++に基づくアーキテクチャである。E-AoSAS++におけるモジュールはアスペクトであることから、次の仮説を提案する。

- 特性層、アトミック特性層のフィーチャは、すべてアスペクトとして扱う。

この仮説に基づいて、プロダクトラインアーキテクチャのコンポーネントとフィーチャとの間の対応関係を、プリンタ制御ソフトウェアを事例として示す。

5.1 特性層、アトミック特性層のフィーチャの対応関係
仮説に基づいて特性層、アトミック特性層のすべてのフィーチャをアスペクトとして扱う場合、ひとつのフィーチャがひとつのアーキテクチャのコンポーネントと対応するが、コンポーネントの数が増加し、アーキテクチャの構造が複雑化してしまう。以下で特性層、アトミック特性層のフィーチャのうち、何をアスペクトとして扱うべきか考察をおこなう。

5.1.1 フィーチャ図上で認識できる横断的コンサーン
フィーチャ図ではフィーチャ間の関連が整理されていることから、複数のコンポーネントに横断するコンサーンをフィーチャ図から認識することが可能である。図3の例では印字がインクカートリッジ、インクノズルに横断していることがフィーチャ図上で認識できる。例に挙げた印字をはじめとする横断的コンサーンはアスペクトと

してモジュール化することが適切であると考えられる。

5.1.2 オア、オルタネイティブ、オプションフィーチャ

フィーチャ図上には、選択を表すオア、オルタネイティブ、オプションフィーチャが存在する。フィーチャ図で選択したフィーチャに対応して、プロダクトラインアーキテクチャ上のコンポーネントが選択され、プロダクトアーキテクチャの構造が決定される。オア、オルタネイティブ、オプションフィーチャをモジュール化することで、プロダクトラインアーキテクチャでのコンポーネントの選択が可能となり、フィーチャ選択時のプロダクトアーキテクチャの作成を容易にすることができる。ただし、これらのフィーチャを、ほかのフィーチャと同様にモジュール化した場合に選択であることを明示することができないので、選択であることを示す手段として、対応するコンポーネントのポリシーにステレオタイプ<<SpecificationModel>>を付加する。また、ノートを用いて選択パターンをアーキテクチャ上に記述する。このように記述することで、対応するポリシーが、実行時の協調動作に加えて、選択もおこなっていることをアーキテクチャ上に表現することができる。図4にステレオタイプとノートを用いた例を示す。

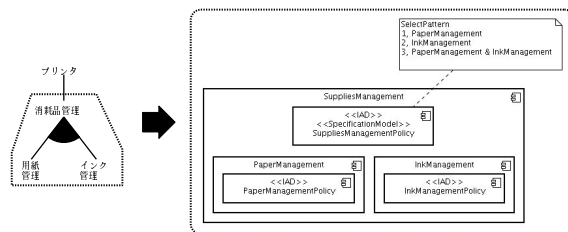


図4 ステレオタイプ、ノートを用いた例

5.1.3 非機能フィーチャ

特性層、アトミック特性層には、機能に関連するフィーチャのほかに非機能フィーチャが記述される。非機能フィーチャを詳細化することによって、フィーチャ図上でほかのフィーチャとの関連を認識することができる。本研究では、非機能フィーチャとして、次のふたつの場合を扱う。

- 関連するフィーチャと対応するコンポーネントの処理を変更することで実現可能なもの
- 実現する際、新たにコンポーネントが必要となるもの

図3における、印字速度、送紙速度のような非機能は、印字、送紙と対応するコンポーネントの処理を変更することで実現可能である。このような場合、非機能フィーチャはアーキテクチャ上にコンポーネントとして記述する必要がない。

図3では、印刷品質を維持するという非機能の実現に、インクヘッドのクリーニング機能が新たに必要とされている。このような場合は、非機能フィーチャの実現に必要な機能フィーチャをモジュール化することが適切である。

また、どちらの場合においても、非機能の実現にハードウェアが必要な場合、ハードウェアが操作環境層にフィーチャとして抽出され、アーキテクチャのコンポーネントと対応する。

5.2 操作環境層のフィーチャの対応関係

操作環境層のフィーチャはプリンタを構成するハードウェア部品であることから、フィーチャとアーキテクチャのコンポーネントが一对一で対応する。

5.3 ドメイン技術層, 実現技術層のフィーチャの対応関係

ドメイン技術層にはドメインに特化した技術, 実現技術層には一般的な規格や開発環境などを記述することから、フィーチャはアーキテクチャのコンポーネントと対応しない。技術の実現に必要なハードウェアがある場合、ハードウェアが操作環境層にフィーチャとして記述され、アーキテクチャのコンポーネントと対応する。

5.4 対応関係に基づくプロダクトラインアーキテクチャ

これまでに整理した対応関係に基づいて作成したプリンタ制御ソフトウェアのプロダクトラインアーキテクチャを図5に示す。

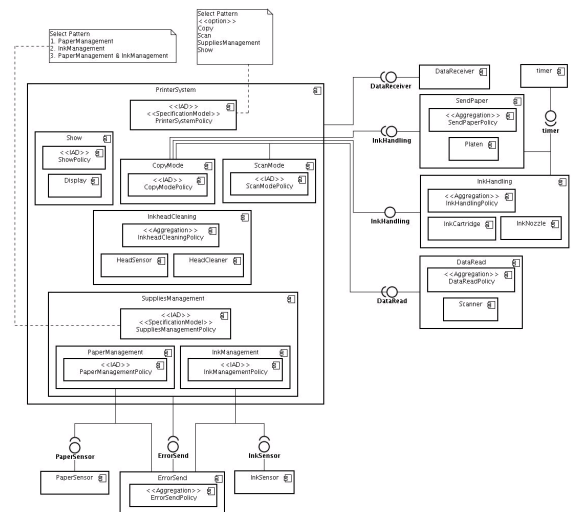


図5 プリンタ制御ソフトウェアのプロダクトラインアーキテクチャ

6 考察

プリンタ制御ソフトウェアを事例として示した対応関係の一般性について、フィーチャ図の各階層毎に考察をおこなう。

6.1 特性層, アトミック特性層の対応関係についての考察

プリンタ制御ソフトウェアを事例とした対応関係で示したように、特性層, アトミック特性層に記述するフィーチャのうち次のものをアスペクトとして扱う。

- オア, オルタネイティブ, オptionalフィーチャ
- フィーチャ図から認識できる横断的コンサーン
- 非機能フィーチャの実現に必要な機能フィーチャ

これらのフィーチャは、ドメインに関わらずフィーチャ図から認識できることから、プリンタと同様の対応関係を適用可能である。

また、プリンタの事例で示した通り、非機能フィーチャに関しては、詳細化の結果、新たに機能フィーチャが必要な場合に、機能フィーチャがコンポーネントと対応する。しかし、詳細化の結果現れたフィーチャを機能として扱うかどうかの明確な基準を定義することは困難であり、最終的な判断は開発者の経験に依存することになる。

6.2 操作環境層の対応関係についての考察

プリンタ制御ソフトウェアに限らず、フィーチャ図の操作環境層には、ハードウェアをフィーチャとして記述することから、プリンタを事例とした対応関係を適用可能である。

6.3 ドメイン技術層, 実現技術層の対応関係についての考察

ドメイン技術層にはドメインに特化した技術, 実現技術層には一般的な規格や開発環境などを記述する。これはプリンタ制御ソフトウェアに限らず共通であることから、プリンタを事例とした対応関係を適用可能である。

7 おわりに

本研究では、プロダクトラインソフトウェア開発における既存仕様モデルの問題点を挙げ、再定義をおこなった。プリンタ制御ソフトウェアについて分析し、フィーチャ図を用いて分析結果の整理をおこなった。プリンタ制御ソフトウェアを事例として対応関係を示し、対応関係の一般性について考察をおこなった。

今後の課題は、より多くのドメインで、ケーススタディをおこない、本研究で提案した対応関係の一般性について検証することである。また、フィーチャ選択時における、アーキテクチャへの変更を自動化するツールの実現が挙げられる。

謝辞

本研究を進めるにあたり、熱心に御指導いただいた沢田篤史教授, 野呂昌満教授, 有益なアドバイスを下さった蜂巢吉成講師, 大学院生の皆様に深く感謝いたします。また、二年間ともががんばってきた沢田研究室, 野呂研究室, 蜂巢研究室のみなさんにも感謝いたします。

参考文献

- [1] K.C.Kang, J.Lee, P.Donohoe, "Feature-Oriented Product Line Engineering", IEEE Software, Vol.19, No.4, p.58-65, 2002.
- [2] L.M.Northrop, "SEI's Software Product Line Tenets", IEEE Software, Vol.19, No.4, p.32-40, 2002.
- [3] Masami Noro, Atsushi Sawada, Yoshinari Hachisu, Masahide Banno, "E-AOSAS++ and its Software Development Environment", Proceedings of the 14th Asia-Pacific Software Engineering Conference(APSEC2007), pp. 206-213, Dec. 2007.