

エディタへの PLSE の適用

2003MT005 荒川 祐衣

2003MT015 日比 千晶

2003MT087 大矢 莉子

指導教員 張 漢明

1 はじめに

エディタはバリエーションが豊富であり、特に系統的な開発を必要としているソフトウェアの一例として考えられる。エディタには編集する対象が多数存在し、また編集対象が同じであっても実行環境が異なるものが存在する。系統的な開発を支援する工学のひとつとして、プロダクトラインソフトウェアエンジニアリング (以下、PLSE) [1] が提案されている。

エディタはバリエーションが豊富にも関わらず製品毎に開発されており、系統的な開発がおこなわれていない。エディタ開発には多大な開発コストと開発時間を費している。

本研究の目的は、テキストエディタのアーキテクチャとフィーチャ図の対応づけをもとにエディタの特徴である編集部分において対応法則を導き出し、再利用可能な部品を抽出する。他のエディタに対応法則を適用し、PLSE 適用可能性について考察する。

本研究は以下のように進める。

- テキストエディタのアーキテクチャの構築
- テキストエディタのユーザの要求分析
- テキストエディタのフィーチャ図とアーキテクチャの対応関係の分析
- テキストエディタの部品の抽出
- エディタの PLSE 適用可能性について考察

2 関連研究

2.1 PLSE

PLSE はソフトウェア開発において、系統的な開発を支援する工学のひとつである。PLSE はソフトウェアファミリとも呼ばれる。PLSE には以下の 3 つの活動がある。

- 核資産開発
- 製品開発
- 管理

核資産開発はドメインエンジニアリングとも呼ばれる。核資産にはアーキテクチャ、ソフトウェア再利用部品、ドメインモデルなどがある。核資産開発はユーザの要求を抽出し、どの要素が再利用出来る共通構造、共通部品であるかなどを分析し、製品を構成する部品の開発をおこなう。製品開発はアプリケーションエンジニアリングとも呼ばれる。核資産開発で開発した部品をもとに製品の開発をおこなう。管理は、開発した部品・製品を再利

用できるように維持することである。

再利用可能な構成要素を部品化し、部品の組み合わせで製品の開発をおこなう。新たに製品開発をおこなう際には、既存の部品を再利用することで効率良く開発を進めることが可能となる。開発サイクルの短い製品においては、PLSE を適用することで製品の生産性は向上し、コストの削減も計れると考えられる。

2.2 FORM

本研究は PLSE を支援する方法論である FORM (Feature-Oriented Reuse Method) [2] の考え方を基盤とする。FORM とは FODA (Feature-Oriented Domain Analysis) [3] から拡張したものである。FORM, FODA ではフィーチャモデリングが提案されていて、フィーチャ図をもちいてユーザの要求分析をおこなう。

2.3 フィーチャ図

フィーチャとは開発者の視点から見たシステムが実現可能なユーザ要求である。フィーチャ図は、ユーザ要求を整理するために必要なフィーチャ同士の関係を見易く記述することができる。

本研究のフィーチャ図は、以下の 4 つの階層に分割してテキストエディタを構成するフィーチャの関連を図示する。テキストエディタのフィーチャの共通性と変更容易性を分析することが可能となり、アーキテクチャを構築する際に利用可能となる。

- 特性層 (Capability Layer)
 - テキストエディタの機能特性や非機能特性を表している
- 操作環境層 (Operating Environment Layer)
 - テキストエディタの操作環境に関する部分を表している
- ドメイン技術層 (Domain Technology Layer)
 - テキストエディタ開発に特化した技術を表している
- 実現技術層 (Implementation Technique Layer)
 - 一般的なソフトウェアなどの実現に利用される技術を表している

2.4 エディタ

エディタとは、編集対象にユーザの入力に応じた編集をするソフトウェアである。エディタにはいくつかの種類がある。その中のグラフィックエディタとテキストエディタを本研究の対象として取り扱う。

- テキストエディタ
 - 文字情報のみのファイル (テキストファイル) の作成・編集・保存をおこなう
- グラフィックエディタ

– 画像ファイルの作成・編集・保存をおこなう

テキストエディタはさらにスクリーンエディタ、ラインエディタ、ストリームエディタの3種類に分けられる。本研究で取り扱うテキストエディタは、マウスカーソルを動かしてテキストの編集をおこなうスクリーンエディタを対象とする。

3 テキストエディタのアーキテクチャの構築

テキストエディタのアーキテクチャを図1に示す。このアーキテクチャは、MVC(Model-View-Controller)の方式を利用して作成した。MVCとは、ソフトウェアの設計モデルの一つで、処理の中核を担う「Model」、表示・出力を司る「View」、入力を受け取ってその内容に応じてViewとModelを制御する「Controller」の3要素の組み合わせでシステムを実装する方式である。

メインの処理はModelにあたるBufferとTextVisitorに実装し、Modelは画面出力などはおこなわない。処理結果はViewに渡され、画面表示などがおこなわれる。ViewはInputDeviceとOutputDeviceである。ユーザからの入力はControllerであるTextEditorが受け取り、何らかの処理が必要な場合はModelに依存し、出力が必要な場合はViewに依頼する。

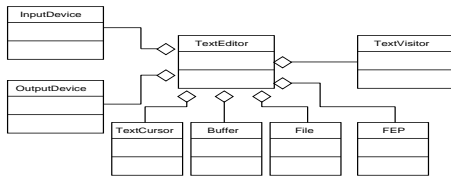


図1 テキストエディタのクラス図

4 テキストエディタのユーザ要求とアーキテクチャの対応づけ

テキストエディタのユーザ要求とアーキテクチャの対応づけをおこなうためには、まず、ユーザの要求分析をする必要がある。本研究では、フィーチャ図を用いてユーザの要求分析をおこなう。

4.1 テキストエディタのフィーチャ図

テキストエディタのフィーチャ図を図2に示す。

編集においては文字入力、切り取り、貼り付け、複製、選択、削除、戻すの7つの機能は必須機能とする。ウィンドウにおいてはテキストカーソルのみが必須機能とされ、任意選択可能な他の機能の中から選択する機能が多い程、操作性が高くなる。

4.2 フィーチャ図とアーキテクチャの対応付け

作成した図2のテキストエディタのフィーチャ図と図1のアーキテクチャの対応付けを図3に示す。

オブジェクト指向とPLSEの観点ではフィーチャとオブジェクトが一对一に対応することが、一方法として考

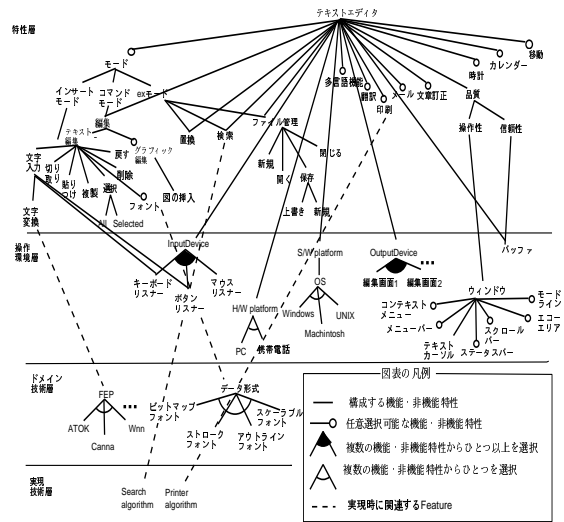


図2 テキストエディタのフィーチャ図

えられる。その考えに基づいてフィーチャとアーキテクチャの対応付けを行った。InputDevice, OutputDeviceはそれぞれに対応し、複製や貼り付け等の編集機能はTextVisitorがVisitorパターンとして機能を管理する。ファイル管理はFile、バッファはBufferと対応する。テキストエディタの特徴であり、テキストエディタに必ず必要なフィーチャのテキストカーソルは、TextCursorと対応する。

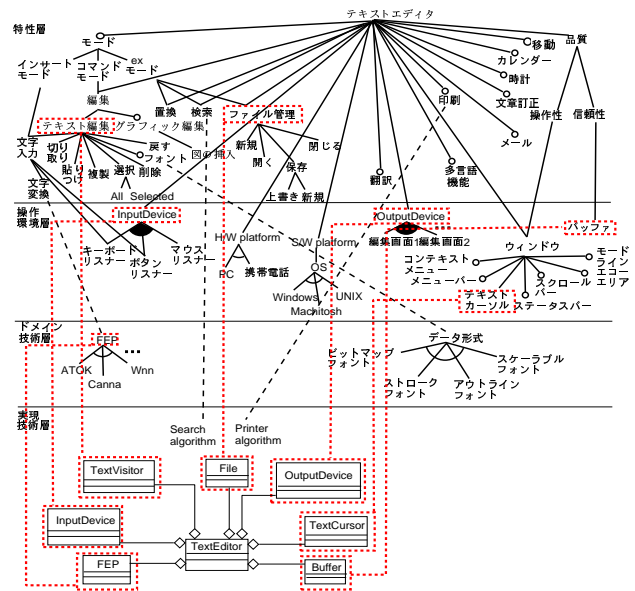


図3 フィーチャ図とアーキテクチャの対応付け

4.3 オプションを選択したテキストエディタ

構築したアーキテクチャにおいて、具体的にオプションを選択した場合を考える。例として、携帯電話のテキストエディタをもちいる。

携帯電話のテキストエディタのフィーチャ図とアーキテクチャの対応付けを図 4 に示す。

本研究で考えたテキストエディタの対応付けと、携帯電話をもちいて考えたテキストエディタの対応付けでは、多くの共通部分を確認した。TextCursor, Buffer, File, FEP, TextVisitor では同様の対応付けとなった。H/Wplatform が携帯電話であるので、CellularPhone が TextEditor を持ち、InputDevice は ButtonListener, OutputDevice は Window となる。アーキテクチャの構築において応用可能性が高いと考えられる。

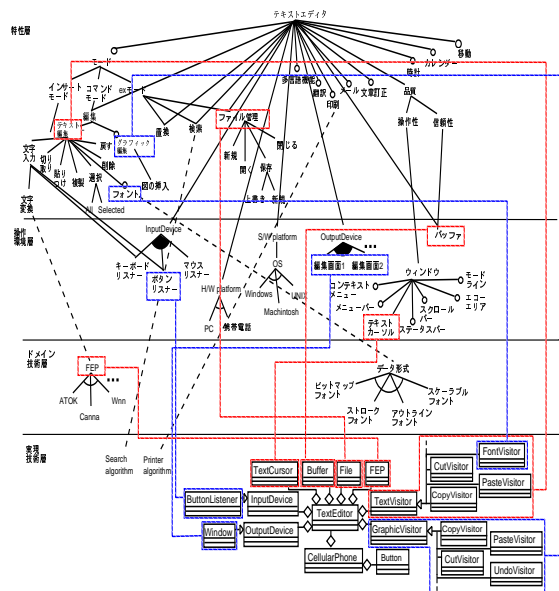


図 4 携帯電話のフィーチャ図とアーキテクチャの対応付け

4.4 Visitor パターン部分の対応法則

テキストエディタのフィーチャ図とアーキテクチャの対応づけから、編集をおこなう Visitor パターン部分の対応法則を導き出した。対応法則を図 5 に示す。

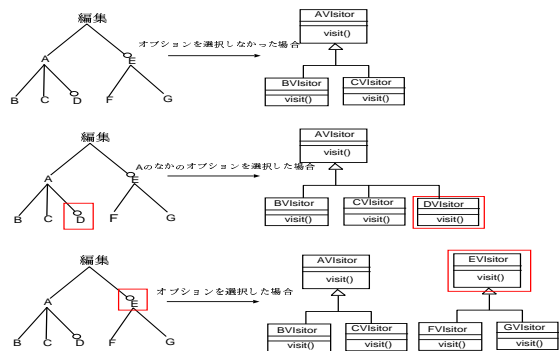


図 5 Visitor パターン部分の対応法則

4.5 テキストエディタの再利用可能な部品

Visitor

テキストエディタには Windows で使用可能なメモ帳と、Linux で使用可能な VI のような種類の違いが存在するが、編集する機能は共通に存在する。

Visitor 部分は実行環境に関係なく再利用できる。テキストエディタでは PC 用でも携帯電話用でもボタンに対する機能の内容は変わらない。機能について違いがあるのは機能数だけである。Visitor が持つサブクラスは変わる可能性があるが、Visitor 自身のメソッドは全く同一のものとなる。Visitor 部分は実行環境に限らず、エディタの種類が同じであれば再利用可能である。

Buffer, File

実際に効果を与える編集対象として Buffer が存在する。Buffer はテキストエディタに必ず存在するので、エディタのもととなる部分として再利用可能である。

ファイルの管理をおこなう File も同様にテキストエディタのもととなる部分であるので、再利用可能である。

TextCursor

テキストエディタにはテキストを入力する部分を示す TextCursor が存在する。切り取りや複製の始点、終点としても必要となる部分で、全てのテキストエディタに存在すると考えられるので、再利用性が高い部品である。

OutputDevice

テキストエディタには必ず編集後をユーザに示す OutputDevice が存在することから OutputDevice ももととなる構造といえる。テキストエディタには、PC 用と携帯電話用のように実行環境が異なる場合や、メモ帳や VI のように種類が異なる場合がある。OutputDevice 自体はテキストエディタに必ず存在するが、実行環境が異なる場合には OutputDevice の種類が異なるために、実行環境が同じ場合のみ再利用性が高い部品である。

InputDevice

入力に関する InputDevice は、必ず存在する部分としてテキストエディタのもととなる構造といえる。

OutputDevice と同様に実行環境が同じ場合のみ再利用性が高い部品と考えられる。

5 考察

テキストエディタのユーザ要求の分析とアーキテクチャの対応づけから導き出した Visitor 部分の対応法則を他のエディタに適用し、再利用可能な部品を抽出する。エディタの PLSE 適用可能性について考察する。

5.1 グラフィックエディタを例とした考察

フィーチャ図とアーキテクチャ対応付けの適用可能性を考察する。テキストエディタを対象として考えたフィーチャ図とアーキテクチャの対応付けを、他のエディタに適用可能であることを示す。グラフィックエディタを例として考察する。

5.1.1 フィーチャ図とアーキテクチャの対応付け

グラフィックエディタのフィーチャ図とアーキテクチャの対応付けを図 6 に示す。

テキストエディタと同様に、ユーザの要求分析としてフィーチャ図を作成する。作成したフィーチャ図とアーキテクチャをテキストエディタの対応付けをもとにグラフィックエディタの対応付けをする。

InputDevice, OutputDevice, File, Buffer の対応はテキストエディタと同じである。編集機能も、ImageVisitor が Visitor パターンとして管理するので、図のような対応になる。テキストエディタとは異なり、テキストカーソル機能が必須ではないために、TextCursor が存在しない。

フィーチャ図とアーキテクチャの対応付けにおいて、テキストエディタとグラフィックエディタでは共通部分が多数確認できた。テキストエディタで進めてきた研究を、グラフィックエディタにも適用可能である。

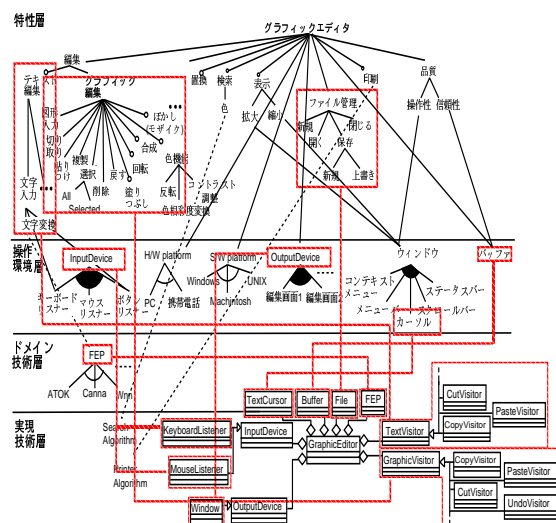


図6 グラフィックエディタのフィーチャ図とアーキテクチャの対応付け

5.2 グラフィックエディタの再利用可能な部品

グラフィックエディタは Windows で使用可能なペイントと、Linux で使用可能な GIMP のような種類がある。テキストエディタと同様に、編集するという機能は必ず存在する。

グラフィックエディタの Visitor も PC 用や携帯電話用のように実行環境が異なっても再利用できる。

実際に効果を与える編集対象である Buffer、ファイルの管理をおこなう File はグラフィックエディタのもととなる構造である。Buffer、File は再利用可能である。また、編集後ユーザに示す OutputDevice や入力に関する InputDevice、はテキストエディタと同様に実行環境が同じ場合のみ再利用性が高い部品となる。

5.3 PLSE 適用可能性

テキストエディタを対象として考えたフィーチャ図とアーキテクチャの対応づけがグラフィックエディタにも適用可能であることがわかった。他のエディタにも適用可能であることがいえる。テキストエディタとグラフィックエディタは構造が似ていて、エディタは構造の再利用が可能である。再利用可能な部品を抽出したことで、PLSE 適用の可能性を確認した。オブジェクト指向と PLSE の観点でフィーチャとアーキテクチャの対応付けを考えたが、構造の面から考えるとこの対応付けには問題が生じると考えられる。

6 おわりに

本研究ではアーキテクチャを構築し、テキストエディタのユーザ要求を分析した。構築したアーキテクチャとユーザ要求との対応づけをおこない、オプションを選択した際のテキストエディタのアーキテクチャとユーザ要求の対応づけを携帯電話のテキストエディタを例におこなった。テキストエディタの部品を抽出した。本研究でおこなったアーキテクチャとユーザ要求との対応づけが他のエディタに適用可能であるかをグラフィックエディタを例に考察した。グラフィックエディタの部品抽出し、エディタの PLSE 適用可能性について考察した。

今後の課題は、アスペクト指向を取り入れ、実際に適用することである。構造が複雑化しているエディタにアスペクト指向を適用し、考察する。構造も考慮した対応付けを考え、実際にソフトウェアに適用する。

謝辞

本研究を進めるにあたり熱心な御指導をいただいた野呂昌満教授、張漢明助教授、有益なアドバイスをいただいた蜂巢吉成講師、大学院生の坂野将秀さん、久松康倫さん、水野耕太さん、石川智子さん、安孫子正康さん、西山遼平さん、太田将吾さん、安江基規さんに深く感謝いたします。また、二年間ともに励まし支え合い頑張ってきた張研究室、野呂研究室一同の方々にも感謝いたします。

参考文献

- [1] L. M. North, "SEI's Software Product Line Tenets" IEEE Softwar, Vol. 19, No. 4, pp. 32-40, July 2002.
- [2] Kyo C.Kang, Sajoong Kim, Jaejoon Lee, Kijoo Kim, Gerard Jounghyum Kim, Euseob Shin, "A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures"
- [3] Kyo C.Kang, "Feature-Oriented Domain Analysis(FODA) Feasibility Study", Nov 1990.
- [4] Erich Gamma, Richard Helm, Ralph Johnson, John Vissides: オブジェクト指向における再利用のためのデザインパターン, 1999.