

アスペクト指向による開発プロセス定義

上野 浩一郎

三菱電機情報技術総合研究所

1. はじめに

近年、ソフトウェア開発の短期化やコスト削減の要求がより強くなり、また開発技術の変化の激しさから、開発プロジェクトのリスクが上がっている。この問題に対する1つのアプローチとして、作業や成果物を標準化した開発プロセスが期待されている。本稿では開発プロセスの定義を容易にし、それを自動実行するためにアスペクト指向技術を導入することを提案する。

2. プロセスに関する従来技術と課題

OMG では開発プロセスに関係するメタレベルを4つに分類している[1]。M1「Process Model」は、具体的な開発プロセスが位置付けられるレベルで、例えば RUP[2]等が該当する。M2「Process Metamodel」は、開発プロセスそのものを表現する構成要素(例えば Activity や WorkProduct)が位置付けられるメタレベルであり、OMG では SPEM[1]として標準化している。M3「MetaObject Facility」は、M2 レベルのインスタンスを表現する構成要素が位置付けられるメタメタレベルである。M0「Performing process」は、具体的なプロジェクトにおいて計画して実行されるインスタンスが位置付けられるレベルである。

次に上述した概念が実際にどのように実践されるのかを図1の従来技術(白抜部分)により説明する。

プロセスエンジニアはプロセスモデリングツール(例えば[2]の Workbench)を用いて、開発におけるアクティビティや成果物などをオブジェクト指向により形式化する。なおプロセスモデリングツールはプロセスメタモデルをスキーマとして用いる。

プロセスエンジニアはプロセスパブリッシュツール(例えば[2]の Builder)を用いて、オブジェクト指向モデルとして形式的に表現されたプロセスモデルから、プロジェクト員が読んで理解できるプロセスコンテンツ(例えば[2]の Web コンテンツ)を作成する。

管理者はプロセスコンテンツを読みながらプロジェクト管理ツール(例えば[3])を用いて、タスクやリソースやコストの計画を作成する。

開発者はプロセスコンテンツを読みながら開発環境を用いて成果物を作成し、作成した成果物を構成管理データとして登録する。

プロジェクト員はプロジェクト管理ツールを用いて計画に対する実績を登録する。

管理者は計画と実績を比較してプロジェクトを管理する。

なおプロジェクトデータと構成管理データが M0 レベルのインスタンスとなる。

プロセスコンテンツは一般に膨大かつ複雑な文書構造になっているため、プロジェクト員にとって理解して実践するには作業負荷が高く、計算機支援のニーズが高い。そこで、プロセスモデルは形式的なので計算機が解釈できることに着目し、プロセスモデルをプロジェクト計画/実行時にも積極的に利用して、プロセス実践を計算機支援することが課題である。

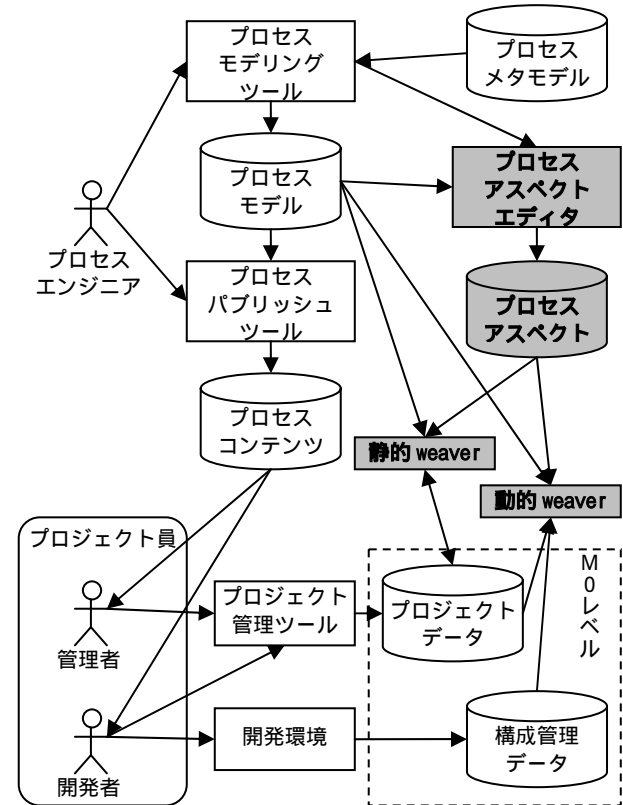


図1 プロセス実践の概念図

注)白抜部分が従来技術、網掛部分が本稿提案技術

3. プロセス定義に関する考察

3.1. 開発パラダイムと開発プロセス

前章で説明した従来技術では、開発プロセスの定義にオブジェクト指向パラダイムを採用している。オブジェクト指向パラダイムは設計や実装と言ったソフトウェア開発技術として既に普及している。これを開発プロセス定義にも援用してパラダイムを統一することは、プロセスエンジニアとプロジェクト員で同じモデリング記法や技法を共用できるので妥当である。

しかし一般論として、オブジェクト指向パラダイ

Aspect Oriented Process Definition

Koichiro UENO

Mitsubishi Electric Corporation

Information Technology R&D Center

ムでは Separation of Concerns[4]の実現に限界があり、具体的にはオブジェクト構造に横断的に現れる Crosscutting Concern を表現するのが困難である。この問題は開発プロセス定義にも現れる。

3.2. プロセスにおける Separation of Concerns

プロセスエンジニアは、ソフトウェア開発の主要工程である要求定義、分析、設計、実装、試験を構成するアクティビティと成果物をオブジェクトとして定義する。これらは Separation of Concerns の用語で言うと Core Concern に該当する。更にソフトウェア開発には上記の主要工程のアクティビティや成果物に直交して共通的に現れる付帯作業(例えば進捗管理や構成管理)が存在する。これらは Crosscutting Concern に該当する。これらの付帯作業を、プロセスエンジニアが従来のオブジェクト指向のみでモデリングする場合、主要工程を構成する多数多様なアクティビティと成果物に対応するオブジェクト毎に、付帯作業の定義を挿入する必要がある。このような付帯作業の定義のしかたは、同じ内容がプロセスモデルの中で分散して重複するので、プロセスエンジニアが定義しそれを保守する負荷が高くなってしまふ。そこで本稿では、開発プロセスにおける Crosscutting Concern をアスペクトとして定義する方法と、更にそれを実行する機能を提案する。

4. プロセスアスペクトとその実行機能の提案

4.1. プロセスアスペクト定義

リスト1と2はプロセスアスペクトを定義した例である。本稿ではアスペクトを記述する言語として便宜上、AspectJ[5]を擬似的に用いる。またリスト中で **BoldItalic** 体で表示している英字名は SPEM が定義している要素であることを示す。リスト1の例は、アクティビティ開始前にその入力成果物を入手するマイルストーンを挿入することを定義している。リスト2の例は、アクティビティ完了後にその出力成果物が構成管理データとして登録されているかを確認することを定義している。

```

1: public aspect 入力成果物入手{
2:   pointcut アクティビティ開始(メタクラス Activity の任意クラス
   anActivity): call(* anActivity.start());
3:   before(メタクラス Activity の任意クラス anActivity) :
   アクティビティ開始(anActivity) {
4:     anActivity が持つ parameterList を取得;
5:     for(parameterList の先頭アイテムから最後尾アイテムまで){
6:       Parameter aParameter =
           parameterList の現在の位置のアイテム;
7:       if(aParameter.kind == input){
8:         WorkProduct aWorkProduct=aParameter.type;
9:         anActivity の先頭マイルストーンとして
           「aWorkProduct の入手」を挿入;
10:      }
11:    }
12:  }
13: }
```

リスト1 プロセスアスペクト例

```

1: public aspect 出力成果物確認{
2:   pointcut アクティビティ完了(メタクラス Activity の任意クラス
   anActivity): call(* anActivity.finish());
3:   after(メタクラス Activity の任意クラス anActivity) :
   アクティビティ完了(anActivity) {
4:     anActivity が持つ parameterList を取得;
5:     for(parameterList の先頭アイテムから最後尾アイテムまで){
6:       Parameter aParameter =
           parameterList の現在の位置のアイテム;
7:       if(aParameter.kind == output){
8:         WorkProduct aWorkProduct=aParameter.type;
9:         構成管理データとして anActivity の出力成果物
           aWorkProduct が登録されているか確認;
10:        if( 登録されていない )警告メール送信;
11:      }
12:    }
13:  }
14: }
```

リスト2 プロセスアスペクト例

4.2. プロセスアスペクトの weaver

前節で説明した方法で定義したプロセスアスペクトを実プロジェクト(MO レベルのインスタンス、即ちプロジェクトデータと構成管理データ)に weaving する機能として2種類の weaver(図1網掛部分参照)を提案する。(1)静的 weaver は、プロジェクト計画時にプロセスアスペクトを解釈し、事前にプロジェクトデータにアスペクトを weaving する機能である。例えばリスト1を実行すると、プロジェクト計画にあるタスクの先頭に適切なマイルストーンを挿入する。(2)動的 weaver は、プロジェクト実行時にプロジェクトデータや構成管理データを監視し、プロセスアスペクトのポイントカットに該当すればアドバイスを実行する機能である。例えばプロジェクト実行時にアクティビティが完了するとリスト2のアドバイスが起動され、出力成果物が登録されているかを確認する。

5. おわりに

本稿では、アスペクト指向技術を導入してプロセスエンジニアのプロセス定義とプロジェクト員のプロセス実践を支援する方法を示した。今後は、プロセスアスペクトを定義するのに適した言語定義と、その実行系の動作検証が課題である。

参考文献

- [1] Software Process Engineering Metamodel: <http://www.omg.org/technology/documents/formal/spem.htm>
- [2] Rational Unified Process: <http://www-6.ibm.com/jp/software/rational/products/pdf/rup.pdf>
- [3] Microsoft Project: <http://www.microsoft.com/japan/office/project/prodinfo/default.msp>
- [4] Walter L. Hursch, Cristina Videira Lopes: Separation of Concerns, Northeastern University, Boston(1995).
- [5] AspectJ: <http://www.eclipse.org/aspectj/>