

BPMN によるソフトウェア開発プロセスの実行環境の提案と評価

壁谷 考洋[†] 伊藤 智基[†] 中道 上^{†‡} 青山 幹雄^{†‡}南山大学 数理情報学部 情報通信学科[†] 南山大学 情報理工学部 ソフトウェア工学科[‡]

1. 研究の背景と課題

オフショア開発などのグローバルソフトウェア開発が進んでいる。そのため、各社の開発プロセスをネットワーク上で連携する必要がある。しかし、開発プロセスの統一的な記述や実行、管理が困難である。本稿では、SOA に基づく開発プロセスの視覚的記述と事前評価を行うためのシミュレータを構築する。

2. 関連研究

SOA (Service-Oriented Architecture) に基づくソフトウェア開発プロセス(SoSD, Service-Oriented Software Development)[4]とその実行環境[2]が提案されている。しかし、開発プロセス管理手法は定義されていない。

3. アプローチ

(1) BPMN の拡張

SoSD モデルの記述を行うため BPMN(Business Process Modeling and Notation)[1]を拡張する。

(2) サービスリポジトリの導入

あるサービスの入出力などサービスの評価に必要な情報が保存可能なサービスリポジトリの導入を行う。

(3) SoSD のシミュレータ

SoSD モデルが満たすべき開発要求や開発プロセスの事前評価が行えるシミュレータを構築する。

4. 開発プロセスの実行状況管理

4.1. 開発プロセスの定義

Task 毎の要求を定義可能にした BPMN を BPMN-S(BPMN for Specification), 要求と実行状況を視覚化した BPMN を BPMN-M(BPMN for Management)と呼ぶ。開発プロセスは複数の Task の集合と捉える。各 Task の要求定義、管理を行うことで、SoSD モデル全体の要求定義、管理を可能にする。

4.2. 実行管理要求の定義

各 Task の要求を記述可能な要素として、BPMN の基本要素であるアノテーションの記述法を定める。アノテーションに記述する要求はコストや、PDM (Precedence Diagramming Method)に基づく開発期間などとする。

4.3. 実行状況の収集

BPMN は BPEL へマッピングが行え、実行できる。しかし、BPEL の実行状況は BPMN に表現できない。

そこで、サービスの入出力などを Web サービス化したサービスリポジトリに保存する。保存する内容は、サービスを実行した開始日、終了日、入出力、プロジェクト毎の ID など実行状況の管理に必要な要素とする。プロジェクト毎の ID の保存により、複数のプロジェクトの実行状況の管理が可能になる。

5. 実行状況の視覚的管理

SoSD モデルを用いて実行状況の管理を視覚的に行うプロセスを図 1 に示す。

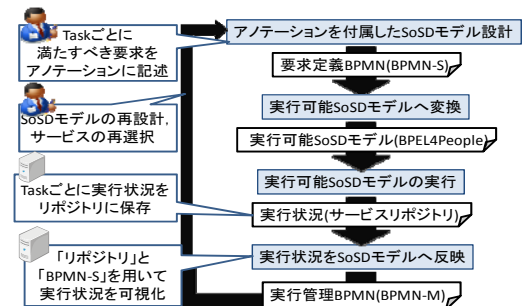


図1 視覚化のプロセス

5.1. BPMN-Sによる開発プロセスの記述

BPMN にソフトウェア開発プロセスを記述する。そして、Task 毎に呼出す開発サービスを決定する。開発サービスとは、開発の各工程をサービスとして提供するものである。開発サービスは呼出し前後にサービスリポジトリを呼び出せるようにインタフェースを付加したサービスとする。こうすることで開発サービスの入出力などの情報を保存できる。呼出し先のサービスに対する要求をアノテーションとして付加する。

5.2. BPMN-Sを実行可能 SoSD モデルへ変換

SOA 基盤上で実行可能な形式に変換する。

5.3. SoSD モデルの実行

BPMN-S で指定した様々な開発サービスを SoSD モデルの実行エンジン上で呼び出す。

5.4. BPMN-M の生成

BPMN-S へ実行状況を表現する。表現方法は、アノテーションへ実行状況の追加、ボトルネックとなる Task の配色を変更する。

6. シミュレータへの要求

6.1. シミュレータの環境

グローバルソフトウェア開発では各社の開発プロセスをネットワーク上で連携できる環境が求められる。シミュレータもネットワーク上で各 Task のシミュレーションを連携できる必要がある。BPMN で開発プロセスを記述した後、シミュレーション用のサービスを呼び出せるようにする。この BPMN を BPEL4People に変換

BPMN-Based Software Process Execution Environment and its Evaluation

[†]Takahiro Kabeya, Tomoki Ito, Department of Information and Telecommunication Engineering, Nanzan University.

^{†‡}Noboru Nakamichi, Mikio Aoyama, Department of Software Engineering, Nanzan University.

することで、シミュレーションが可能になる。

6.2. シミュレーション用サービス

シミュレーション用サービスとは、SoSD のシミュレーションを行うためのサービスである。このサービスは Task への要求、前 Task からの成果物を入力とする。また、要求に対する検証結果、次の Task 担当者による確認のために必要な成果物を出力とする。

7. プロトタイプの実装と実行

7.1. 実装する目的

- (1) BPMN-S の妥当性の確認
- (2) BPMN-M の妥当性の確認
- (3) シミュレータの有効性の確認

以上 3 項目を確認することを実装の目的とする。

7.2. プロトタイプの実装

プロトタイプのシステム構成図を図 2 に示す。BPMN の記述には Intalio[3]を用いた。BPMN-M 生成ツールは Java で実装した。規模は 430 行である。サービスリポジトリは MySQL を用いて Java で実装し、Web サービス化した。規模は 100 行である。

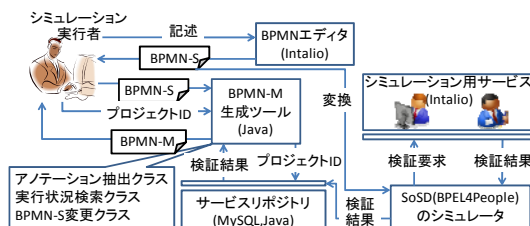


図 2 システム構成図

7.3. プロトタイプの実行

以下にシミュレータの構築、BPMN-M 生成ツールの実行手順を示す。

(1) BPMN-S の設計

SoSD モデルを BPMN-S として記述した (図 3)。

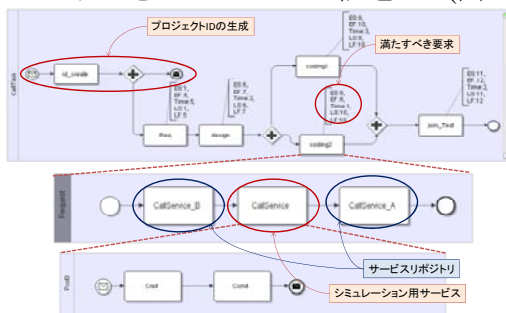


図 3 BPMN-S

要求として納期に PDM の概念を適用した。アノテーションには所要時間, ES, EF, LS, LF を記述することで、各 Task の開発期間を定義した。

(2) SoSD モデルの変換と実行

BPMN-S を BPEL4People へ変換し、実行エンジンを用いることにより、開発プロセスを実行する。

(3) BPMN-M の生成

BPMN-S に記述されている Task 毎の納期と実績、シミュレーションの結果を獲得する。これらの情報を

BPMN-M 生成ツールに入力することで BPMN-M を生成した。シミュレーションの結果、納期を超えた Task を表示した BPMN-M(図 4)を示す。また、LF を超過した Task を表示した BPMN-M(図 5)を示す。

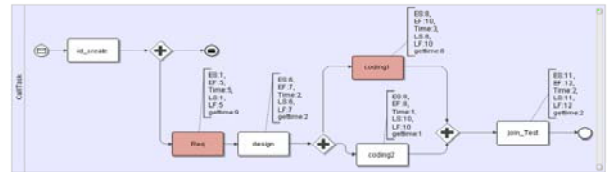


図 4 ボトルネックになり得る Task

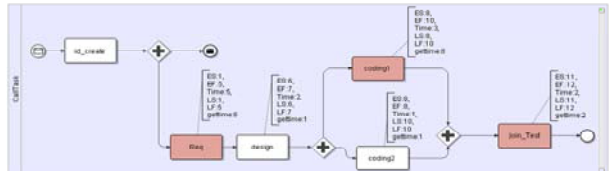


図 5 LF を超過した Task

8. 評価

8.1. ネットワークを介したグローバルな分散開発

ソフトウェア開発のサービスを標準的なインタフェース記述言語によって定義した。ソフトウェア開発プロセスをサービスの連携として、統一的なモデルで記述可能になった。

8.2. ソフトウェア開発プロセスの実行と管理

BPMN-S の利用により、各 Task への要求を明確に表現し、複数の Task の集合であるプロジェクトの要求を明確に表現可能になった。BPMN-M の利用により、開発プロセスのボトルネックを視覚的に表現し、開発プロセス改善の支援が可能になった。

8.3. 開発の委託と管理

SoSD のシミュレータを構築した。これにより、実際にオフショア開発などで外部委託を行う開発プロセスの事前評価が可能になった。

9. まとめ

BPMN を Task 毎に要求を定義できるように拡張した BPMN-S、要求と実行状況の差を表現できるように拡張した BPMN-M を提案した。これらの拡張により、SoSD モデルの要求、ボトルネックが表現可能になった。また、SoSD のシミュレータを構築、実行した。これを用いて開発プロセスの事前評価、改善点の視覚的表現を行い、提案手法の有効性を評価した。

参考文献

- [1] A. Agrawal, et al., Business Process Modeling and Notation 2.0 (BPMN 2.0), OMG, Jul. 2010, <http://www.omg.org/spec/BPMN/2.0>.
- [2] 浅岡 奈津貴, ほか, ソフトウェア開発モデルとその実行環境の提案, 情報処理学会第 167 回ソフトウェア工学研究会, Mar. 2010, pp. 1-8.
- [3] Intalio, Japan Community, <http://oss-bpms.jp/>.
- [4] 大原 晋吾, ほか, サービス指向に基づくソフトウェア開発モデル化方法論の提案, 情報処理学会第 163 回ソフトウェア工学研究会, Mar. 2009, pp. 249-256.