

SOAに基づくソフトウェア開発プロセスの提案

2007MI079 伊藤 智基 2007MI087 壁谷 考洋

指導教員 青山 幹雄

1. はじめに

近年、オフショア開発やアウトソーシングの利用によるグローバルソフトウェア開発が進んでいる。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する必要がある。しかし、統一的な開発プロセスの記述や実行、管理が困難である。

本稿では、SOAに基づく開発プロセスの管理を行うための提案、事前評価を行うためのシミュレータの構築を行う。

2. 研究課題

グローバルソフトウェア開発の課題を示す。

(1) ネットワークを介したグローバルな分散開発

ソフトウェア開発プロセスはネットワーク上で連携する事を想定していない。よってネットワークを介したグローバルソフトウェア開発に対応したプロセスモデルが必要である。

(2) 統一的なソフトウェア開発プロセスの実行と管理

複数の組織が開発に関わり、階層構造を形成している。また、組織毎に開発プロセスが異なるため、組織間での連携コストが増大している。このことから、組織間でのプロセスの階層的な記述と管理が可能なプロセスモデルが必要である。

(3) 開発の委託と管理

オフショア開発やアウトソーシングでは委託する以前に、明確な仕様の決定、タスクの明確化、詳細なスケジューリングが必要になる。そのため、事前評価と開発の支援を行う環境が求められている。

3. 関連研究

SOA (Service-Oriented Architecture)[5]に基づくソフトウェア開発プロセス[6]とその実行環境[3]が提案されている。

前者ではSOAの三階層に基づいたソフトウェア開発プロセスのモデル化の方法が提案されている。ソフトウェア開発をサービスの集合と捉え、ソフトウェア開発は開発サービスの実行により実現される。しかし、プロセスの管理方法は定義されていない。

後者ではSoSD (Service-Oriented Software Development) と呼ぶ実行可能なプロセスへの変換方法が提案されている。ソフトウェア開発プロセスをSOA基盤上で実行可能な記述に変換する。しかし、SoSDの実行状況の管理は定義されていない。

4. アプローチ

(1) サービス指向ソフトウェア開発

自組織内でSoSDのシミュレーションが行える環境を構築する。シミュレーション環境構築の際、SoSDモデルの設計にBPMN(Business Process Modeling and Notation)[1]を用いることで抽象度の高い表現でSoSDのシミュレーションが可能になる。

このSoSDモデルの実行にはBPEL4People(Business Process Execution Language for People)を用いる。BPEL4Peopleとは、BPEL(Business Process Execution Language)を拡張し、人手の作業を追加したWebサービスのプロセス実行言語である(図1)。シミュレーションを行うことで、SoSDモデルが満たすべき要求や開発プロセスの事前評価を可能とする。

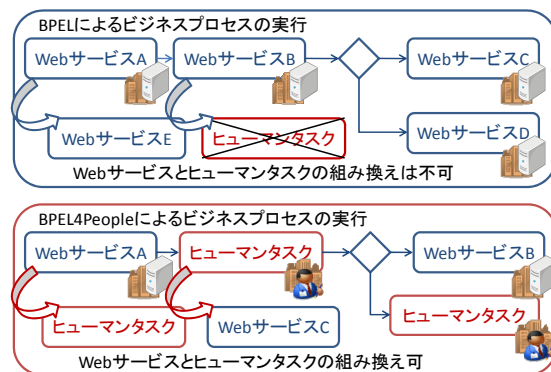


図1 サービスの組換え

(2) サービスリポジトリの導入

実行可能プロセスで呼び出すサービスの入出力などを保存可能なサービスリポジトリの導入を行う。利用するサービスの実行状況をサービスリポジトリに保存することで、サービスの評価が可能になる。このサービスリポジトリをSoSD実行中は実行状態の管理、SoSDモデル設計中は開発サービスを選択する際の指標として利用する。

(3) BPMNの拡張

本稿では、SoSDモデルの記述のためにBPMNを拡張する。Task毎に満たすべき要求を定義できるように拡張したBPMNをBPMN-S(BPMN for Specification)、要求と実行状況の差を視覚化したBPMNをBPMN-M(BPMN for Management)と呼ぶ。SoSDモデルを用いて開発プロセス

の実行状況の管理が行えると、各 Task が満たすべき要求の確認、ボトルネックの発見が視覚的に可能になる。

視覚的にボトルネックを表現することで、モデル中の改善点すべきが明確になる。

5. 開発プロセスの実行状況管理

5.1. 開発プロセスの定義

BPMN はワークフローによってビジネスプロセスの設計と管理を行うためのプロセス表記法である。BPMN の特徴として BPEL へマッピング可能なことが挙げられる。大規模、複雑化したソフトウェア開発では、階層化した組織の開発プロセスを統一的な形式で記述する必要がある。SoSD モデルに拡張した BPMN を適用するため、プロセスのサービス化が可能になる。サービス化すると、開発プロセスの詳細を隠蔽でき、階層的な記述が可能になる(図 2)。

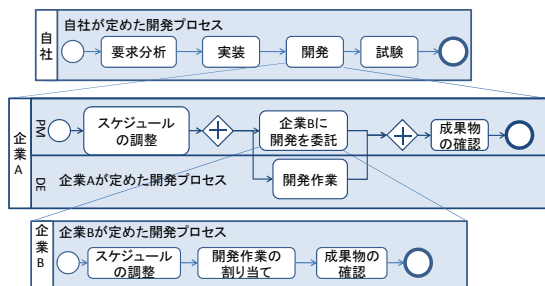


図 2 開発プロセスの階層構造表現

5.2. 実行管理要求の定義

BPMN の基本要素であるアノテーションの記述方法は特に定められていない。そこで、各 Task が満たすべき要求を記述可能な要素として、アノテーションの記述方法を定める(図 3)。アノテーションに記述する要素は PDM(Precedence Diagramming Method)の概念を用いた Task 毎の納期、委託先と意志疎通を図るために共通フレーム 2007 や ISO12207 で定められているプロセス名などとする。

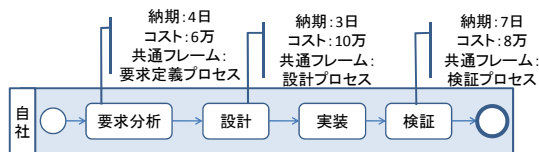


図 3 アノテーションによる Task の要求記述

5.3. 実行状況の収集

BPMN から BPEL, BPEL4People へマッピング可能であっても、実行状況をそのまま 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(図 4)。サービスリポジトリに保存する内容は、サービスの実行を開始した日程、終了した日程、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 を保存することで、複数のプ

ロジェクトの実行状況管理が可能になる。

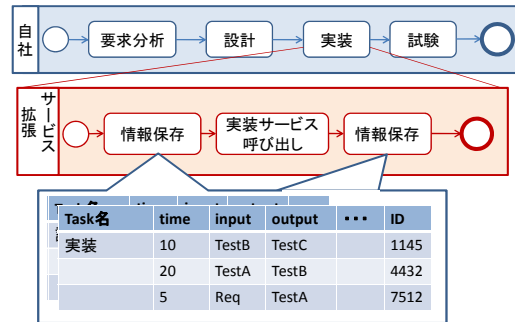


図 4 サービスリポジトリへのインターフェース付加

6. 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 5 に示す。

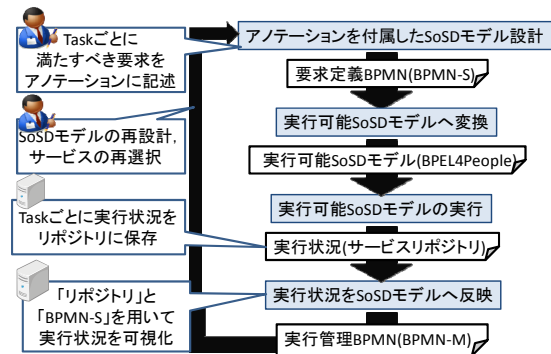


図 5 視覚化のプロセス

6.1. BPMN-S による開発プロセスの記述

BPMN-S を用いてソフトウェア開発プロセスを記述する。そして、Task 毎に呼び出す開発サービスを決定する。

開発サービスとは、各開発プロセスをサービスとして提供するものである。開発サービスを呼出す時、サービスリポジトリのインターフェースを呼出し前後に付随する。これにより、インターフェースに開発サービスの入出力などの情報を入力し、保存を行う。利用するサービス決定後、Task 毎に呼出し先のサービスが満たすべき要求をアノテーションとして付加する。

6.2. BPMN-S を実行可能 SoSD モデルへ変換

BPMN-S を SOA 基盤上で実行可能 SoSD モデルへ変換する。この実行可能 SoSD モデルは BPEL4People を指す。

6.3. SoSD モデルの実行

SoSD モデルを実行エンジン上で実行する。BPMN-S で指定した様々な開発サービスを呼び出し、ソフトウェア開発サービスを実行する。

6.4. BPMN-M の生成

サービスリポジトリに保存した実行状況を用いて、

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，ボトルネックになり得る Task の配色変更とする。

7. シミュレータの構築

SoSD のシミュレータへの要求を以下に示す。

7.1. シミュレーション環境

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する環境が求められる。そこで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する必要がある(図6)。

SoSD モデルの記述に BPMN を用いる。BPMN で開発プロセスを記述した後，呼び出すサービスを指定する。呼び出すサービスはシミュレーション用に構築したサービスとする。この BPMN を BPEL4People に変換する。BPEL4People に変換することで，ネットワークを介して分散的なシミュレーションが可能になる。

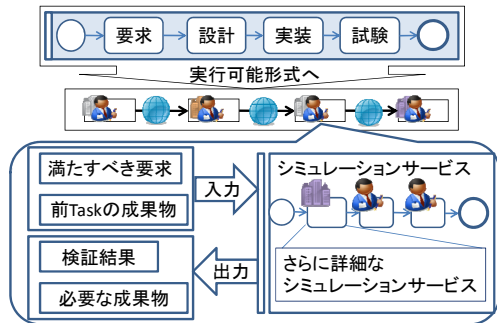


図 6 シミュレーション環境

7.2. シミュレーション用サービス

SoSD のシミュレータは各 Task が指定したコストや納期通りに実行可能か確認することが目的である。そこで，各 Task の担当者が要求された納期などが妥当であるか確認を行えるサービスが求められる。このサービスは入力を Task が満たすべき要求，前 Task からの成果物とする。出力は満たすべき要求に対する検証結果，次の Task 担当者が確認を行うために必要な成果物とする。また，開発プロセスは複数の Task の集合と捉える。各 Task のシミュレーションを行うことで，開発プロセス全体のシミュレーションを可能にする。

8. プロトタイプの実装と実行

8.1. 目的

プロトタイプを実装する目的を以下に示す。

- (1) BPMN-S の妥当性の確認
アノテーションに要求定義が行えているか確認する。
- (2) シミュレータの有効性の確認
満たすべき要求の事前評価が行えるか確認する。
- (3) BPMN-M の妥当性の確認
ボトルネックの視覚化が行えているか確認する。視覚化

した情報により開発プロセスの管理が可能か確認する。

8.2. プロトタイプの実装

プロトタイプの実装構成図を図 7，生成ツールの処理の流れを図 8 に示す。BPMN の記述には Intalio[4]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。

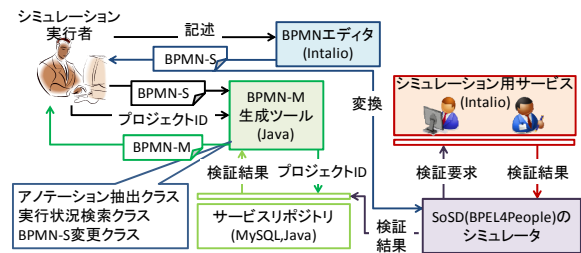


図 7 システム構成図

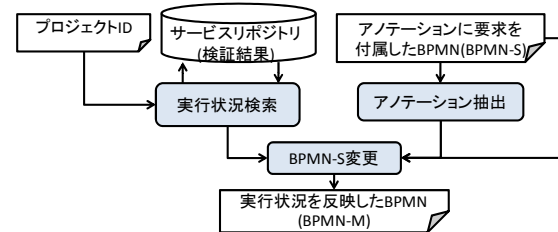


図 8 生成ツールの処理の流れ

8.3. プロトタイプの実行

シミュレータの実行と BPMN-M 生成ツールの実行を行う。シミュレータの SoSD モデルは BPMN-S で記述する。BPMN-S で記述することで，満たすべき要求の獲得が可能になる。これをシミュレーション結果と重ね合わせて，BPMN-M を生成する。

(1) BPMN-S の設計

SoSD モデルを BPMN-S として記述する(図9)。

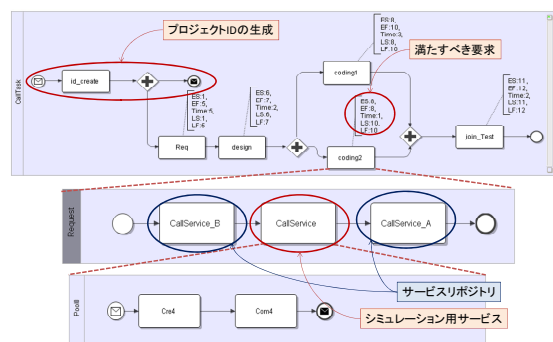


図 9 BPMN-S モデル

プロトタイプでは各 Task が満たすべき要求を納期に限定した。この納期に PDM の概念を適用し，所要時間，

ES(Earliest Start time), EF(Earliest Finish time), LS(Latest Start time), LF(Latest Finish time)をアノテーションに記述した。このように記述することで、Task 単体の遅れ、開発プロセス全体への影響が認識可能になる。

(2) SoSD モデルへ変換と実行

BPMN-S を BPEL4People へ変換し、そのインタフェースとなる WSDL を生成する。この実行可能 SoSD モデルである BPEL4People を Intalio のワークフローエンジンを用いることで、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 毎に満たすべき納期と実績、シミュレーションを行った結果を獲得する。これらの情報を BPMN-M 生成ツールに入力することで BPMN-M を生成する。

シミュレーションの結果、納期を超えた Task を明示した BPMN-M の表示例(図 10)を示す。また、納期内で実行可能だが、設定した最も遅い日程である LF を超過した Task を視覚化した BPMN-M の表示例(図 11)を示す。

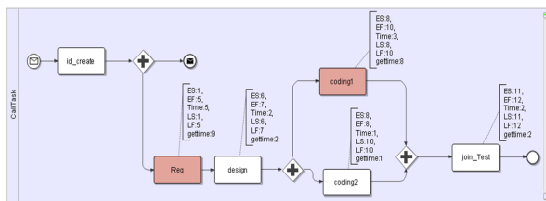


図 10 ボトルネックになり得る Task

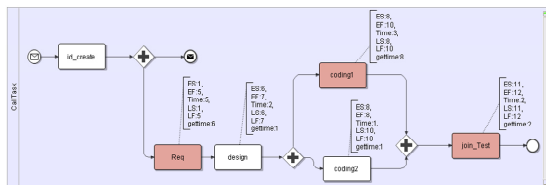


図 11 LF を超過した Task

9. 評価

研究課題に対する評価を行う。

9.1. 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。

9.2. 統一的なソフトウェア開発プロセスの実行と管理

BPMN を拡張して視覚的に要求と実行状況の管理が可能な BPMN-S と BPMN-M を提案した。BPMN-S を用いることで、各 Task が満たすべき要求を明確に表現可能になった。これにより、複数の Task の集合であるプロジェクトの要求を明確に表すことが可能になった。BPMN-S をプロジェクト単位で開発プロセスに対する要求の定義が可能なモデルと捉えた。

BPMN-M を用いることで、開発プロセス中のボトルネックが視覚的に表現可能になった。これにより、開発プロセス改善の支援が可能になった。

9.3. 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。

これは事前評価と支援を行う環境が必要という課題を解決している。

10. 今後の課題

本稿では、開発に対する要求として納期のボトルネックを可視化した。しかし、ボトルネックは納期以外でも発生する。納期以外の要求と実行状況の違いも表現する必要がある。また、アノテーションに PDM の概念を用いて、所要時間、ES, EF, LS, LF の記述を行った。しかし、PDM では Task 毎の所有時間が分かれば、これらは計算可能である。

今後、同期ゲートウェイや排他ゲートウェイを考慮に入れ、計算方法を提案する必要がある。

11. まとめ

本稿では、SoSD モデルの記述のために BPMN を拡張した。Task 毎に満たすべき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、改善点が明確に表現可能になった。

また、シミュレータを構築、実行した。構築の際、開発プロセスの記述に BPMN-S を用いて開発工程の要求定義を行った。シミュレーション結果と開発工程に対する要求を生成ツールで重ね合わせることで BPMN-M を生成した。これら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、提案の有効性の評価した。

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] T. T. Allweyer (ed.) BPMN 2.0-Introduction to the Standard for Business Process Modeling: Bod, 2010.
- [3] 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.
- [4] Intalio, Japan Community, <http://oss-bpms.jp/>.
- [5] D. Krafzig, et al. Enterprise SOA, Prentice Hall, 2005.
- [6] 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.