

# 実行可能サービス指向ソフトウェアプロセスの提案と評価

浅岡 奈津貴<sup>†</sup> 中道 上<sup>‡</sup> 青山 幹雄<sup>‡</sup>

南山大学大学院 数理情報研究科<sup>†</sup> 南山大学 情報理工学部 ソフトウェア工学科<sup>‡</sup>

## 1. 背景と問題点

現在, ソフトウェア開発はグローバル化が進み, 複数の組織がネットワークを介して協調して開発を行う複雑な形態となっている. そのため, ソフトウェア開発の統一的な実行と管理が困難である. 本稿は, ソフトウェア開発をサービスの複合体と考え, サービス指向基盤上で統一的にソフトウェア開発を実行するための実行可能な記述のモデルと変換方法を提案する.

## 2. 関連研究

ソフトウェア開発プロセスの形式的な記述としてプロセスプログラミング[5]が研究されている. しかし, ネットワークを介した連携は扱われていない. また, サービス指向ソフトウェア開発モデルが提案されている[4]. しかし実行可能なサービス指向ソフトウェア開発プロセス記述は未定義である.

### 2.1. サービス指向ソフトウェア開発 (SoSD)

ソフトウェア開発では, 要求分析などの人手の作業と開発ツールなどのコンピュータ処理が連携している. 人手の作業とコンピュータによる処理をサービスと捉え, サービスの組み合わせと実行で実現する開発をサービス指向ソフトウェア開発 (Service-oriented Software Development: SoSD) と呼ぶ. サービスはプラットフォームとは独立に定義され, 統一的なインタフェースを提供する.

### 2.2. 実行可能なビジネスプロセス記述言語

サービス指向基盤上で実行可能なサービス連携の記述形式に BPEL (Business Process Execution Language) がある. また, BPEL に人手の作業を組み込む拡張仕様として, BPEL4People (WS-BPEL Extension for People)[1]と WS-Human Task (Web Services Human Task)[2]が策定されている.

## 3. 実行可能なビジネスプロセス記述

### 3.1. 変換プロセス

ソフトウェア開発プロセスから実行可能な記述への変換の流れを図 1 に示す.



図 1 サービス指向ソフトウェア開発 (SoSD)

SoSD に基づくプロセス記述 (SoSD モデル) は, BPMN (Business Process Model and Notation) で記述し, 実行可能な記述 (実行可能 SoSD モデル) へ変換する. SoSD モデルを実行形式とは独立に定義することで, 複数の組織で統一的に記述可能となる. さらに, BPMN で記述することによって視覚的なプロセス定義が可能になる.

### 3.2. ソフトウェア開発サービスと管理モデル

#### 3.2.1. サービスの定義

SoSD ではサービスのインタフェースはプロセスの成果物となる. サービスとして, 以下の特性が求められる.

- (1) 可観測性 (Observable): 契約などの条件下で外部に公開可能であり, サービスの外部から観測可能である. 開発者や組織間で, 成果物として合意が取れている.
- (2) 検証可能性 (Verifiable): サービスの入出力である成果物がサービスの仕様に従っていることの正当性を検証可能である.

これらの特性を満たした成果物を入出力とする作業や処理がサービスとなり得る.

#### 3.2.2. 開発サービスと管理サービスの分離

ソフトウェア開発は, 開発と管理からなる. 管理には, 進捗管理, リソース管理, コスト管理のようにプロセスによらない汎用的管理がある. ここで汎用的管理を管理サービス, ソフトウェア開発のプロセスを開発サービスと呼ぶ.

SoSD モデルでは開発サービスのみを記述し, 開発サービスと管理サービスを分離する. 開発サービスには常に管理サービスが関連付けられる. 管理サービスは SoSD モデルから実行可能な記述への変換時に開発サービスに埋め込む.

#### 3.2.3. 開発サービスの管理方法

管理サービスが進捗などの情報を取得するために用いるインタフェースをメタインタフェースとして開発サービスに定義する (図 2). メタインタフェースは, サービスの管理情報を取得す

Executable Service-Oriented Software Process Description.

<sup>†</sup> Natsuki Asaoka, Graduate School of Mathematical Science and Information Engineering, Nanzan University.

<sup>‡</sup> Noboru Nakamichi, Mikio Aoyama, Department of Software Engineering, Nanzan University.

するためのインタフェースを返す。

プロジェクトマネージャ(PM)は管理サービスが提供するインタフェースを用いて開発サービスの管理情報を取得する。取得した情報を分析し、進捗管理、コスト管理、リソース管理を行う。この3つの管理活動は並行に行われる。

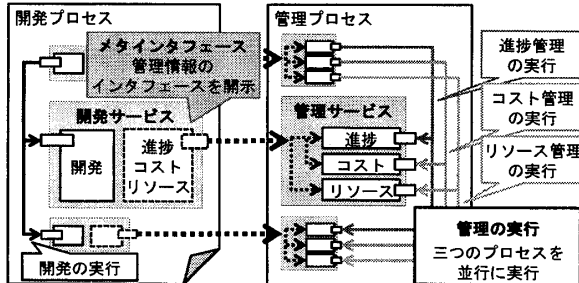


図2 開発サービスの管理モデル

### 3.3. モデルに基づく変換とツール

#### 3.3.1. 変換方法

実行可能 SoSD モデルは BPEL4People と WS-Human Task で記述する。

- (1) 開発サービスのみで構成された SoSD モデルを BPMN で記述する。
- (2) BPMN2BPEL plug-in[3] により BPMN から BPEL に変換する。
- (3) (2)の BPEL に管理サービスを付加する。
- (4) 人手で実行するサービスを WS-Human Task に基づき Task として定義する。また、BPEL4People プロセスから Task を呼び出すインタフェースをサービス呼び出し情報として記述する。(3)の管理サービスを付加した BPEL に、Task 定義とサービス呼び出し情報を加え、実行可能 SoSD モデルに変換する。

変換の流れを図3に示す。

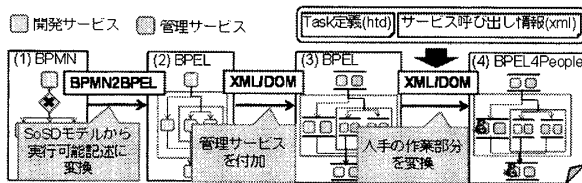


図3 実行可能 SoSD モデルへの変換

#### 3.3.2. 変換ツール

- (1) BPMN から BPEL への変換: 以下のソフトウェアを用いて変換する。
  - 1) Eclipse 3.4.2
  - 2) BPMN2BPEL eclipse plug-in
- (2) 開発サービス(BPEL)に管理サービスを付加: 開発サービス(BPEL)に、並行に実行する管理サービスを付加する。並行処理を記述する flow 要素を作成し、開発サービスと新しく作成した管理サービスを flow 要素の子要素とする。この変換を Java と XML/DOM 変換を用

いて実装した。

- (3) 人手の作業を置き換え BPEL4People に変換: BPEL4People は BPEL の extensionActivity 要素を用いて人手の作業を記述する。人手の作業をこの要素で置き換え、子要素に人手の作業のサービス情報を記述した peopleActivity 要素を加える。peopleActivity 要素は Task 定義とサービス呼び出し情報から生成する。この変換を Java と XML/DOM 変換を用いて実装した。

### 4. 例題による評価

ソフトウェア開発作業内の下記 2 つの作業をサービスと定義し、SoSD モデルを記述した。

- (1) 試験計画の変更
- (2) コード変更と単体試験

この SoSD モデルを実行可能な SoSD モデルに変換した(図4)。実行可能 SoSD モデルは、BPEL4People の実行エンジン(ActiveBPEL など)を用いて実行可能である。

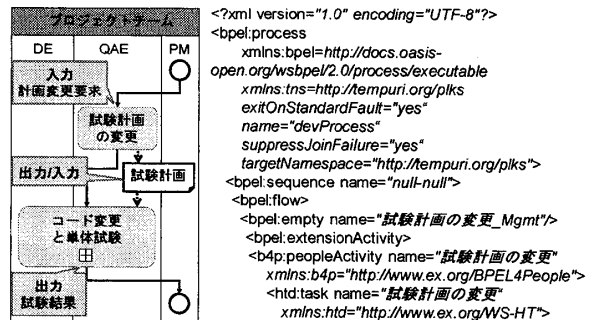


図4 SoSD モデルと実行可能 SoSD モデル

### 5. まとめと今後の課題

本稿ではソフトウェア開発をサービスの集合体と捉え SoSD モデルとして BPMN で記述し、BPEL4People と WS-Human Task を用いた実行可能な SoSD モデルへの変換方法を提案した。また、管理サービスを用いた連携による進捗管理、コスト管理、リソース管理の実現方法を提案した。

今後、管理サービスを連携した実行可能な管理プロセスの生成方法を検討する。

### 6. 参考文献

- [1] A. Agrawal, et al., WS-BPEL Extension for People (BPEL4People), Ver. 1.0, 2007.
- [2] A. Agrawal, et al., Web Services Human Task (WS-HumanTask), Ver. 1.0, 2007.
- [3] Lgbanuelos, bpmn2bpel, Jun. 2008, <http://code.google.com/p/bpmn2bpel/>.
- [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, ソフトウェア工学研究会, Mar. 2009, pp. 249-256.
- [5] L. Osterweil, Software Processes are Software too, Proc. ICSE 1987, IEEE Computer Society, Mar. 1987, pp. 2-13.