

ソフトウェア開発プロセスのサービスモデルとその実行環境の提案

M2008MM003 浅岡 奈津貴

指導教員: 青山 幹雄

1. はじめに

ソフトウェア開発は、ソフトウェアの大規模化に伴い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な分散化が進んでいる。複数の組織がネットワークを介して協調し、ソフトウェア開発は複雑な形態になっている。そのため、ソフトウェア開発プロジェクトの統一的な実行と管理が困難である。

本稿では、ソフトウェア開発にサービス指向を適用して開発をサービスの組み合わせと捉え、ソフトウェア開発プロセスをモデル化する。さらに、開発プロセスをサービス指向基盤上で実行可能な記述に変換する方法を提案する。

2. 背景と問題点

大規模化する開発の特徴と問題点を以下にあげる。

- (1) ネットワークを介したグローバルな分散開発
開発に必要なリソースやコストが増加し、ソフトウェア開発の様々な作業を外部に委託するオフショア開発などが行われている。このような開発では、複数の異なる組織がネットワークを介して協調して開発を進めている。しかし、これまでのソフトウェア開発プロセスではネットワーク上での連携は想定されていない。
- (2) 統一的なソフトウェア開発プロセスの実行と管理
ソフトウェア開発の大規模化により、複数の組織が開発に関わり、階層的な構造をつくる。また、組織ごとに異なる開発モデルと管理モデルを持つため、組織間での連携のコストが増大している。統一的なプロセスのモデル化とネットワーク上での実行管理が求められている。

3. 関連研究

本研究と関連する研究を以下に挙げる。

3.1. ソフトウェアプロセスに関する研究

プログラミング技術を用いたソフトウェア開発プロセスの記述が提案されている。ソフトウェア開発プロセスをプログラミングし、プログラムを実行することでソフトウェア開発プロセスを実行する。しかし、ネットワークを介した連携は扱われていない。

3.2. ソフトウェア開発にサービス指向を適用した開発モデルの提案

SOA 三階層に基づいたソフトウェア開発のモデル化が提案されている。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、実行可能なソフトウェア開発プロセスの具体的な記述の定義がない。

3.3. 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。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) と

WS-Human Task (Web Services Human Task)が策定されている。これにより、人手の作業と Web サービスの両方を含んだ SOA 基盤上で実行可能なビジネスプロセスの記述が可能となる。BPEL4People に沿ってビジネスプロセスを記述し、WS-Human Task では人手の作業をヒューマンタスクとしてサービス化して記述する。

4. アプローチ

4.1. 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。

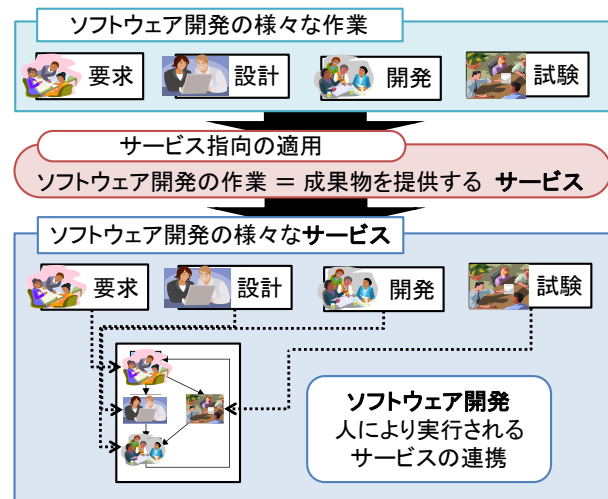


図 1 サービス指向ソフトウェア開発

4.2. 実行可能なビジネスプロセス記述への変換

本稿ではソフトウェア開発プロセスの定義とその実行可能な記述への変換という2段階に分けて考える。

SoSD に基づいて作成されたソフトウェア開発プロセス (SoSD モデル)をサービス指向コンピューティング基盤上で実行可能な記述(実行可能 SoSD モデル)に変換する。変換したプロセスをサービス指向コンピューティング基盤上で実行することで、ネットワーク上でのソフトウェア開発の実行を実現する。

汎用的管理を行うための管理モデルの導入

ソフトウェア開発は開発と管理から成り立っており、管理には進捗管理、リソース管理、コスト管理のように開発のプロセスによらない汎用的管理が存在する。開発には常に管理が関連付けられる。ここで、開発作業を開発サービス、汎用的管理を管理サービスと呼ぶ。

本稿では、管理サービスが開発サービスを管理する仕組みを管理モデルとして定義し、管理サービスの連携によって管理プロセスを作成する。

5. サービス指向ソフトウェア開発(SoSD)

5.1. 変換の流れ

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 2 に示す。

- (1) SoSD モデルの設計
- (2) SoSD モデルを実行可能 SoSD モデルへの変換
- (3) 実行可能 SoSD モデルの実行

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り、技術者以外でも SoSD モデルのプロセス定義に参画できる。

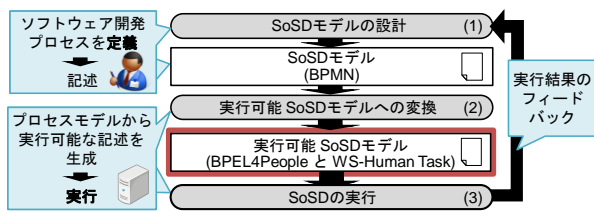


図 2 変換の流れ

5.2. サービス定義

5.2.1. Web サービスとソフトウェア開発サービスの比較

SOA/Web サービスの条件[3]のうちの 7 項目について、ソフトウェア開発サービスに対応付ける(表 1)。この結果から、ソフトウェア開発サービスが満たすべき条件を明確にし、ソフトウェア開発プロセスを構成するソフトウェア開発サービスの定義を行う。

表 1 Web サービスの条件とソフトウェア開発サービス

条件	SOA/Web サービス	ソフトウェア開発サービス
再利用性	再利用可能となる意味のある機能を提供	ソフトウェア開発において意味のある機能を提供
インタフェースが契約により定義	サービスのエンドポイント、オペレーション、入出力メッセージによりインタフェースを定義	エンドポイント(サービスを提供する組織)、オペレーション、サービスの入出力によりインタフェースを定義
疎結合	サービス間に依存関係がなく、独立して実行可能	ソフトウェア開発サービス間に依存関係がなく、独立して実行可能
実装やロジックの隠蔽	インタフェースのみで相互運用可能。Web サービスによって実装の隠蔽	インタフェースのみで相互運用可能。サービス内の作業の詳細や開発環境が隠蔽されるように設計
組み合わせ可能	より価値の高いサービスとなるよう他のサービスと組み合わせ可能	ソフトウェア開発サービスを組み合わせることでソフトウェア開発プロセスが実現可能
自律	サービスの実行は自身で完結し、他から実行の支援なしに実行可能	特定の組織内で実行が完結し、他のサービスの支援なしに実行可能

5.2.2. サービスの定義

ソフトウェア開発において独立して実行可能であり、まとまった作業単位となって開発工程を隠蔽できるものを開発サービスとする。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する。サービスのインタフェースとなる成果物には以下の 2 つの特性が求められる。

- (1) 可観測性(Observable)

サービスのインタフェース情報に含まれる成果物は、契約などの条件下で外部に公開可能である。

- (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という正当性を検証可能な必要がある。

5.3. 管理モデル

5.3.1. 管理サービスによる開発サービスの管理

ソフトウェア開発サービスの管理を行うためには、開発サービスが提供するインタフェース情報のみでは不十分である。そこで、開発サービスに内部情報を提供するためのメタインタフェースを追加する。管理サービスはメタインタフェースを用いて、開発サービスがもつ進捗取得インタフェース、コスト取得インタフェース、リソースインタフェースを取得し、それらのインタフェースから得られる情報を利用して汎用的管理を行う。

これにより、管理サービスは特定の開発サービスに依存しない統一的方法で管理情報を取得できる。

また、管理サービスは開発サービスと並行に実行するので、開発のプロセスに沿った管理が可能になる。

5.3.2. 管理サービスの連携による管理プロセスの実現

プロセスベースの管理ではプロセスの進行とともに管理を行うので、管理対象ごとに管理プロセスが定義できる。本稿では管理サービスを連携させて管理プロセスを作成する。管理サービスの管理対象は進捗、コスト、リソースの 3 つであり、管理プロセスの数は 3 つとなる。

また、管理サービスは前に実行されたプロセスの予定と実績を入力とし、管理対象のプロセスの予定と実績を出力とする。管理プロセスは予定と実績の情報を連携することでソフトウェア開発の管理を実現する。

管理サービスの連携によって、ソフトウェア開発プロセスの管理を行うサービス指向ソフトウェア開発管理モデルを図 1 に示す。

管理モデルを利用することで、メタインタフェースを用いてソフトウェア開発を統一的に管理することが可能となる。

また、開発サービスが提供する管理情報取得インタフェースを追加することで管理対象を増やし、その管理プロセスを生成することが可能である。

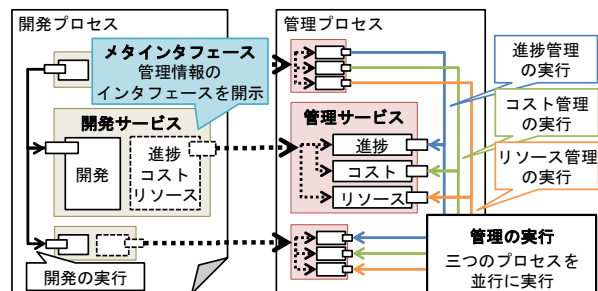


図 1 開発サービスの管理モデル

6. 実行可能 SoSD モデルへの変換

6.1. 変換方法

実行可能 SoSD モデルは BPEL4People と WS-Human Task で記述する。変換の流れを図 2 に示す。

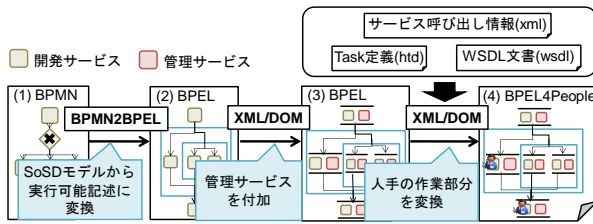


図 2 実行可能 SoSD モデルへの変換

(1) SoSD モデルの記述

開発サービスを定義し、BPMN でプロセスを記述する。

(2) BPMN から BPEL への変換

変換は Eclipse3.4.2 と BPMN2BPEL plug-in エラー! 参照元が見つかりません。を用いる。

(3) BPEL に管理サービスを付加

開発サービスと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。

(4) BPEL から実行可能 SoSD モデルへの変換

人手で実行するサービスを定義した WS-Human Task 文書とプロセスからサービスを呼び出すタイプを記述したサービス呼び出し情報、サービスのインタフェースを定義する WSDL 文書の 3 つ管理サービスを付加した BPEL に加え、BPEL4People へ変換する。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 WS-Human Task 文書とサービス呼び出し情報から生成する。

6.2. プロトタイプ

プロトタイプとして作成した変換ツールの処理の流れを図 3 に示す。変換ツールは Java を用いて作成し、開発規模は 3 クラス 599 行となった。

(1) 管理サービスの付加

入力として受け取った BPEL に、管理サービスを付加した BPEL を出力する。この変換は MgmtServiceCreator として実装した。

(2) 実行可能プロセスへの変換

BPEL 文書、サービス定義、インタフェース情報、サービス呼び出し文書を入力として受け取り、BPEL を BPEL4People へ変換して出力する。

プロセス連携に人手の作業で実現されるサービスが含まれる場合には、人手の作業をヒューマンタスクとして定義する WS-Human Task 文書と定義したヒューマンタスクのインタフェース情報が記述された WSDL 文書を入力として必要とする。また、プロセス連携に Web サービスで実現されるサービスが含まれる場合には、Web サービスのインタフェース情報が記述された WSDL 文書を入力として必要とする。さらに、これらのサービス定義がプロセスに固有かどうかにより、プロ

セスからサービスを呼び出す方法が異なる。そこで、呼び出し方法を記述したサービス呼び出し文書を入力として与える。この変換を Java と XML/DOM 変換を用いて Bpel2b4p として実装した。

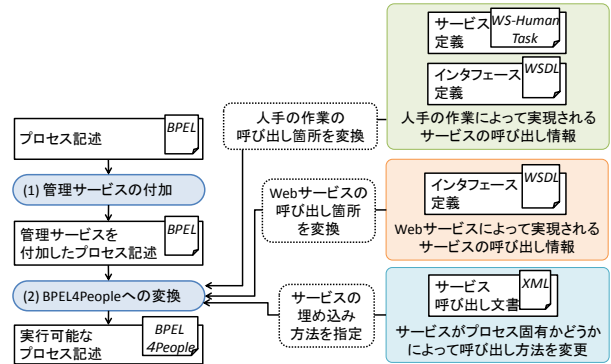


図 3 変換ツールの処理の流れ

7. 例題

ソフトウェア開発プロセスモデリングのための例題[7]のプロセスを参考に、コード変更とその関連するモジュールの単体試験作業に対して SoSD を適用する。

7.1. コード変更と単体試験プロセス

このプロセスは、(1) コード変更要求、(2) コード変更、(3) コンパイル、(4) 単体試験の 4 つのアクティビティから構成される。

7.2. SoSD の適用

各アクティビティの入出力のうち、サービスとして求められる特性を満たした成果物を明確にし、SoSD モデルを BPMN として記述する。各アクティビティの成果物を表 2 に、それらのアクティビティをサービスと捉えてプロセスを記述した SoSD モデルを図 4 に示す。

表 2 アクティビティの成果物

アクティビティ	入力	出力
コード変更要求	変更許可	変更要求
コード変更	変更要求	ソースコード
コンパイル	ソースコード	オブジェクトコード
単体試験	オブジェクトコード	試験結果

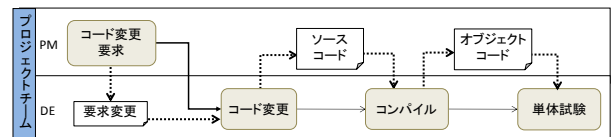


図 4 コード変更と単体試験プロセスの SoSD モデル

記述した SoSD モデルの BPMN 記述から BPEL を生成し、ツールを用いて管理サービスを付加する。

次に、人が行う作業をサービスとして WS-Human Task 文書で定義し、プロセスからサービスを呼び出す呼び出し情報を作成する。この例ではコード変更要求、コード変更、単体試験が人手で行う作業とした。人手の作業をサービスとして定義した WS-Human Task 文書の一部を図 7 に示す。

また、コンパイルは Web サービスで実現されているとし、

その WSDL を取得する。

```
<htd:humanInteractions...>
<htd:logicalPeopleGroups>
  <htd:logicalPeopleGroup name="プロジェクトマネージャ"></htd:logicalPeopleGroup>
  ...
</htd:logicalPeopleGroups>

<htd:tasks>
<htd:task name="コード変更要求">
  <htd:interface
portType="tns:CodeChangeRequestPT" operation="request"
responsePortType="tns:CodeChangeRequestCallbackPT" responseOperation="requestResponse"/>
<htd:peopleAssignments>
  <htd:potentialOwners>
    <htd:from logicalPeopleGroup="プロジェクトマネージャ"></htd:from>
  </htd:potentialOwners>
</htd:peopleAssignments>
  ...
</htd:task>
  ...
</htd:tasks>
</htd:humanInteractions>
```

図 7 WS-Human Task 文書の一部

これらの情報を入力としてツールを用いて BPEL を BPEL4People に変換した(図 8)。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<bpel:process name="CodechangeAndUnittest"...>
  <bpel:sequence name="コード変更要求-単体試験">
    <bpel:flow>
      <bpel:empty name="コード変更要求_Mgmt"/>
      <bpel:extensionActivity>
        <b4p:peopleActivity name="コード変更要求"
xmlns:b4p="http://www.example.org/BPEL4People">
          <htd:task name="コード変更要求" xmlns:htd="http://www.example.org/WS-HT">
            <htd:interface operation="request"
portType="tns:CodeChangeRequestPT"
responseOperation="requestResponse"
responsePortType="tns:CodeChangeRequestCallbackPT"/>
            <htd:peopleAssignments>
              <htd:potentialOwners>
                <htd:from logicalPeopleGroup="プロジェクトマネージャ"/>
              </htd:potentialOwners>
            </htd:peopleAssignments>
            ...
          </htd:task>
        </b4p:peopleActivity>
      </bpel:extensionActivity>
    </bpel:flow>
    ...
  <bpel:flow>
    <bpel:empty name="コンパイル_Mgmt"/>
    <bpel:empty name="コンパイル"/>
  </bpel:flow>
  ...
</bpel:sequence>

  <b4p:humanInteractions xmlns:b4p="http://www.example.org/BPEL4People">
    <htd:logicalPeopleGroups xmlns:htd="http://www.example.org/WS-HT">
      <htd:logicalPeopleGroup name="プロジェクトマネージャ"/>
      ...
    </htd:logicalPeopleGroups>
  </b4p:humanInteractions>
</bpel:process>
```

図 8 実行可能 SoSD モデル

8. 評価

8.1. SoSD に基づくソフトウェア開発プロセスの妥当性

SoSD を用いて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、ソフトウェア開発プロセスをサービスの連携という統一的なモデルで記述することが可能になる。

8.2. 実行可能 SoSD モデルへの変換方法の妥当性

プロトタイプを用いて BPEL4People と WS-Human Task に基づいた、SOA 基盤上でネットワークを介したソフトウェア開発プロセスを実行可能な記述に変換できることを確認した。

8.3. 管理モデル導入の妥当性

付加した管理サービスを連携させることにより、ソフトウェア開発プロセスと並行に実行される管理プロセスを生成可能な管理モデルを提案した。この管理モデルによって、管理サービスはメタインタフェースを用いて統一的な方法で開発サ

ービスの管理が可能になる。

8.4. プロトタイプの妥当性

プロトタイプでは、管理サービスの付加を行ったあとでサービスをヒューマンタスクまたは Web サービスに置き換えることで、管理サービスも開発サービスと同様に変換の行うことが可能である。

また、今回 WS-Human Task 文書の作成のためのサービス定義と記述を人が行っているが、今後ツールの支援が望まれると考えられる。

9. 今後の課題

BPMN で記述しているスイムレーンの情報などを BPEL4People に利用できていない。そこで、スイムレーン情報によってヒューマンタスクの作業者を決定するなど、BPMN から BPEL4People と WS-Human Task への変換を支援するツールが必要である。

取得した汎用的管理の情報を用いて、開発サービスに変更を加える方法について今後検討する必要がある。また、管理プロセスの生成方法は定義していない。今後、実行可能な管理プロセスの生成方法を検討する。

10. まとめ

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、開発サービスに汎用的管理を行う管理サービスを関連付け、管理サービスの連携によって進捗管理、コスト管理、リソース管理などの管理プロセスを実現する管理モデルを提案した。

参考文献

- [1] A. Agrawal, et al., WS-BPEL Extension for People (BPEL4People), Ver. 1.0, Jun. 2007.
- [2] A. Agrawal, et al., Web Services Human Task (WS-Human Task), Ver. 1.0, Jun. 2007.
- [3] T. Erl, Service-Oriented Architecture, Prentice Hall, 2005.
- [4] M. I. Kellner et. al., Software Process Modeling Example Problem, Proc. of 6th ISPW, Oct. 1990, IEEE Computer Society, pp. 19-29, [篠田 陽一ほか(訳), ソフトウェアプロセスモデリングのための例題, 情報処理, Vol. 36, No. 5, May 1995, pp. 385-391].
- [5] Lgbanuelos, bpmn2bpel, Jun. 2008, <http://code.google.com/p/bpmn2bpel/>.
- [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, 第 163 回ソフトウェア工学研究会, Mar. 2009, pp. 249-256.
- [7] L. Osterweil, Software Processes are Software too, Proc. ICSE 1987, IEEE Computer Society, Mar. 1987, pp. 2-13.