

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

大原 晋吾[†] 中道 上[‡] 青山 幹雄[‡]

[†]南山大学大学院 数理情報研究科 数理情報専攻 [‡]南山大学 数理情報学部 情報通信学科

本稿では、ソフトウェア開発を、ソフトウェア開発を実行するサービスの協調から成る、サービスの複合体と考え、サービス指向に基づく統一的なモデル化の方法を提案する。さらに、ビジネスプロセスマネジメント(BPM)のフレームワークに基づき、ソフトウェア開発のモデル化から実行、管理を行うサービス指向ソフトウェア開発モデル化プロセスを提案する。また、例題を用いて、ソフトウェア開発プロセスのモデリングを行い、提案したモデル化方法論の妥当性を示す。

Service-Oriented Modeling Methodology for Software Development

Shingo Ohara[†], Noboru Nakamichi[‡], Mikio Aoyama[‡]

[†]Graduate School of Mathematical Sciences and Information Engineering, Nanzan University

[‡]Dept. of Information and Telecommunication Engineering, Nanzan University

In the article, we regard that software development is composed of services of software development, and propose a unified Service-Oriented modeling method for software development. We propose Service-Oriented modeling process for software development based on framework of Business Process Management (BPM). We prove the proposed modeling method by describing example problems.

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、オフショア開発の利用など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する必要があり、ソフトウェア開発は複雑な形態になっている。そのため、統一的なソフトウェア開発プロジェクトの実行と管理が困難である。

本稿では、ソフトウェア開発をサービスの複合体と考え[2]、ソフトウェア開発におけるサービス指向の適用を提案する。従来のビジネスにおける SOA と BPM のモデルと対応する、ソフトウェア開発におけるモデルを提案することにより、ソフトウェア開発における SOA と BPM の適用を明確にする。さらに、対応するモデルに基づいて、ソフトウェア開発のモデル化から実行、管理を実現するサービス指向ソフトウェア開発モデル化プロセスを提案する。また、ソフトウェア開発プロセスの例題の記述と分析を通して、提案するモデル化方法論の妥当性を示す。

2. ソフトウェア開発の問題点

ソフトウェアの大規模化に伴う、近年のソフトウェア開発の特徴を以下に説明する。

(1) ソフトウェア開発の分散化

ソフトウェア開発は、コスト削減を期待したオフショア開発や、アウトソーシングの利用など、グローバルな分散化が進んでいる[5]。そのため、複数の異なる組織がネットワークを利用し、作業の協調によってソフトウェア

開発を実行している(図 1)。

(2) 多様な作業と成果物

開発工程は分割され、複数の組織に割り当てられる。各作業の成果物として、設計書やソースコードなど、多様な成果物が混在している。

(3) ソフトウェア開発の階層化

ソフトウェア開発の大規模化に対応し、ソフトウェア開発を行う組織は階層的な構造になっている。



図 1 ソフトウェア開発の現状

これらの特徴から、ソフトウェア開発は複雑な形態であるため、ソフトウェア開発プロジェクトを統一的に実行、管理することが困難になっている。効率的なソフトウェア開発の実行のために、複数の階層的な組織間で協調する作業を統一的に管理、実行する必要がある。

3. 関連研究

本稿と関連する研究として、以下の二つがある。

(1) ソフトウェアプロセスに関する研究[13]

プログラミング技術によるソフトウェア開発プロセスの記述が提案されている。本研究では、ソフトウェア開発に対して、サービス指向の概念を適用したソフトウェア開発のモデル化プロセスと、サービスの協調としてのソ

ソフトウェア開発のモデル化を提案する。

(2) SOA と BPM に関する研究[10]

自動車産業において、ビジネスプロセスモデルから SOA 上のサービスへのマッピングが提案されている。本稿では、ソフトウェア開発におけるサービスとソフトウェア開発プロセスの関連を明確にすることで、ソフトウェア開発における SOA と BPM の適用を表したモデルを提案する。

4. SOA に基づくソフトウェア開発

4.1. ソフトウェア開発におけるサービス指向の適用

現在、ビジネスにおいて BPM と SOA を用いたビジネスの実現が行われている[4, 8]。ビジネスをサービスの集合として捉え、モデル化したビジネスプロセスモデルに基づき、サービスの協調としてビジネスを実行する。さらに、その実行をモニタリングすることでビジネスの継続的な改善を可能とする。

この一連の BPM と SOA によるビジネスの実現方法の、ソフトウェア開発における適用を提案する(図 2)。以下の 4 点からビジネスとソフトウェア開発を対応付けることで適用を行う。

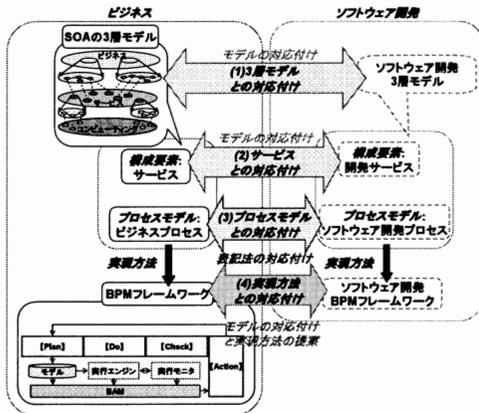


図 2 ソフトウェア開発におけるサービス指向の適用

(1) 3 層モデルとの対応付け

ビジネスでは、サービスとビジネスプロセスの関連を示す SOA の 3 層モデル[1]が提案されている。SOA の 3 層モデルにソフトウェア開発の要素を対応付け、ソフトウェア開発プロセスとサービス、サービスの構成要素の関連を明確に示す。

(2) サービスとの対応付け

ビジネスでは、サービスの特性である疎結合によって、ビジネスを実現する柔軟性の高いシステムが構築可能となる。従来の SOA におけるサービスの条件と比較することで、ソフトウェア開発サービスの条件を明確にする。

(3) プロセスモデルとの対応付け

ビジネスでは、ビジネスをモデル化したビジネスプロセスに基づいて、サービスの協調を実現する。ソフトウ

ェア開発も、開発プロジェクトの作業順序をモデル化したソフトウェア開発プロセスに基づいて実行される。ビジネス領域で用いられるプロセス表記法である BPMN をソフトウェア開発に対して意味拡張する。

(4) 実現方法との対応付け

ビジネス領域では、BPM のフレームワークに基づいた継続的なビジネスの実行、管理が可能である。そこで、ソフトウェア開発を、ソフトウェア開発プロセスに基づいたサービスの協調として実現するために、ソフトウェア開発の BPM フレームワークを提案する。また、このフレームワークに基づいてサービス指向に基づくソフトウェア開発モデル化プロセスを提案する。

4.2. ソフトウェア開発の SOA3 層モデル

SOA とは、ソフトウェアシステムをサービスの組み合わせとして捉えるシステム構築の概念である。ビジネスとサービスの関連は、ビジネス、サービス、コンピューティングの 3 層から成るモデルとして定義される[1]。本稿では、従来の 3 層モデルとソフトウェア開発の要素を対応づけることでソフトウェア開発における SOA3 層モデルを提案する。

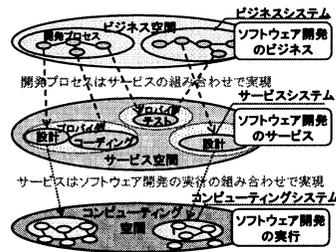


図 3 ソフトウェア開発の SOA3 層モデル

(1) ソフトウェア開発ビジネス

ビジネスシステムは、ソフトウェア開発プロセスを表したモデルと対応付ける。

(2) ソフトウェア開発サービス

サービスシステムは、ソフトウェア開発に含まれる作業と対応付ける。

(3) ソフトウェア開発の実行

コンピューティングシステムは、ヒューマンタスクや開発支援環境と対応づける。

上位のソフトウェア開発プロセスは、異なる組織が提供する、ソフトウェア開発サービスの組み合わせで実現する。ソフトウェア開発サービスは、ソフトウェア開発の実行の要素を組み合わせることで実現される。

4.3. ソフトウェア開発のサービス化

従来の SOA/Web サービスにおけるサービスの条件[3]とソフトウェア開発サービスを比較し、ソフトウェア開発サービスの条件を明確にする。表 1 にその対応を示す。また、サービスの条件の比較から、ソフトウェア開発サービスが満たすべき条件を説明する。

表 1 サービスの比較

条件	SOA/Web サービス	ソフトウェア開発サービス
再利用性	ある機能を実行するオペレーションの集まりで再利用可能となる意味のある機能を提供する。	ソフトウェア開発において意味のある機能を提供(開発工程単位、ソフトウェアツールの利用、など)する。
インタフェースにより定義	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、サービスのルールによりインタフェースが定義される。	サービスの入出力、エンドポイント(サービスを提供する組織)、サービス品質によりインタフェースが定義される。
疎結合	SOA の本質的特性である。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。特に Web サービスによって実装の隠蔽、下位のミドルウェアや OS が隠蔽されるように設計されている。	インタフェースのみで相互運用可能。下位のソフトウェア開発の実行(ヒューマンタスク、開発支援環境など)が隠蔽されるように設計されている。
組合せ可能	他のサービスと組み合わせ、より価値の高いサービスとなるよう組合せ可能であること。	ソフトウェア開発サービスを組み合わせることによって一連のソフトウェア開発プロセスが実現可能。
自律	サービスはそれ自体で実行が完結しており、他から実行の支援がなくても実行可能。	ソフトウェア開発サービスは特定の組織内で実行が完結している、まとまった機能を提供する。
発見可能	サービスが公開され、かつ、サービスの決定性を損なうことなく、一意に決定可能。	ソフトウェア開発サービスはサービスレポジトリに公開され、決定性を損なうことなく、一意に決定可能。

(1) ソフトウェア開発サービスの単位

ソフトウェア開発サービスは、ソフトウェア開発において意味と独立性を持った単位で提供される必要がある。そこで、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である開発工程をソフトウェア開発のサービス単位と考える。各工程は、データなどの成果物を生成する。この成果物を、ソフトウェア開発サービスのアウトプットと考える。

また、ソフトウェア開発は、人による作業によって実行され、開発支援ソフトウェアを利用する。この開発支援ソフトウェアを、汎用で独立した細粒度の機能を提供するサービス(開発支援ソフトウェアサービス)として考える。ソフトウェア開発サービスはヒューマンタスクと開発支援ソフトウェアサービスから成る複合サービスである(図 4)。

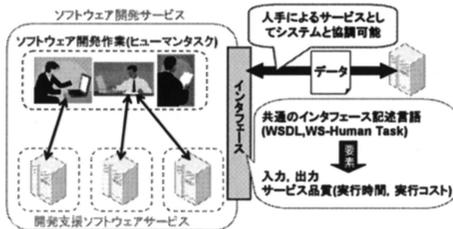


図 4 ソフトウェア開発サービス

(2) ソフトウェア開発サービスのインタフェース

開発プロジェクトをサービスの協調として統一的に実行、管理するためには、人手によって実行されるサービスも、ソフトウェアによって実行されるサービスと同様に、協調が可能なインタフェースモデルが必要である。

また、ソフトウェア開発プロジェクトにおいて、プロジェクトの納期やコストは非常に重要な要素であるため、ソフトウェア開発サービスに対する、厳しい時間とコストの制約が考えられる。ソフトウェア開発サービスがインタフェースで定義する要素として、サービスの品質を表す実行時間、実行コストが必要である(図 4)。

(3) ソフトウェア開発サービスの発見

ソフトウェア開発サービスは、人的リソースによって実行する。サービスの実行可能性や品質は、リソースのスケジュールに応じて変化するため、サービスコンシューマは利用時のサービスの品質に基づいてサービスを発見する必要がある。また、ソフトウェア開発サービスの実行時間は、従来のサービスに比べて長寿命(ロングライフサイクル)になるため、サービスの実行が正常に完了することがプロジェクトの実行において重要である。サービスコンシューマは以上の要素を考慮し、サービスを発見、決定する必要がある。

4.4. ソフトウェア開発の BPM フレームワーク

サービス指向に基づくソフトウェア開発を BPM フレームワークに基づいて実現する。図 5 に提案するソフトウェア開発に適用した BPM のフレームワークを示す。

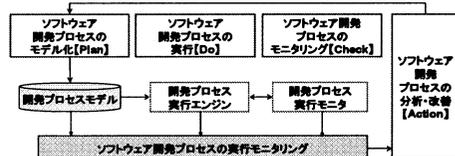


図 5 ソフトウェア開発の BPM フレームワーク

(1) ソフトウェア開発プロセスのモデル化

ソフトウェア開発を、サービスの協調として実現するために、ソフトウェア開発プロセスをモデル化する必要がある。ソフトウェア開発プロセスを、高水準な表現が可能で、視覚的に理解しやすい表記法を用いてモデル化する。

(2) ソフトウェア開発プロセスの実行

ソフトウェア開発プロセスとソフトウェア開発サービスを対応づけ、プロセス実行エンジンで実行可能言語であるプロセス記述言語に変換する。開発プロセス実行エンジンによってソフトウェア開発プロセスを実行、管理する。

(3) ソフトウェア開発プロセスのモニタリングと改善

開発プロセス実行モニタでリアルタイムに実行をモニタリングし、迅速なソフトウェア開発プロセスの分析と改善を行う。ソフトウェア開発において重要な要素を KPI として設定し、モニタリングすることでソフトウェア開発の管理を行う。さらに、そのモニタリングデータに基づき、ソフトウェア開発プロセスの改善を実現する。

4.5. ソフトウェア開発モデル

サービス指向に基づくソフトウェア開発では、以下の要素から成る開発モデルが考えられる(図 6)。

れる、プロセス実行エンジンで実行可能なプロセスモデルを表す。

- (12) 開発プロセス実行エンジン: 実行可能モデルを実行する実行エンジンを表す
- (13) 開発プロセス実行モニタ: ソフトウェア開発プロセスの実行をモニタリングするモニタを表す。

5.5. モデル化プロセスの工程

サービス指向ソフトウェア開発モデル化プロセスの工程は以下に従う。

(1) ソフトウェア開発プロセスモデルの設計

ソフトウェア開発プロセスは、開発工程を実行するソフトウェア開発サービスの協調としてモデル化される。ソフトウェア開発プロセスモデルの設計では、プロセスフレームワークで定義されているプロセス全体の構造に対して、サービスライブラリで公開されているソフトウェア開発サービスを組み合わせる。

(2) プロジェクトプロセスモデルの設計

プロジェクトプロセスモデルの設計では、プロジェクト固有の要求を満たす各サービスを発見、評価、決定する(図 9)。

1) ソフトウェア開発サービスの発見

ソフトウェア開発サービスの発見は、開発サービスプロバイダのプロバイダ情報に基づいて行われる。プロバイダ情報には、そのプロバイダの経営情報などが含まれる。サービスコンシューマはこの情報に基づいて、開発サービスプロバイダを評価する。

2) ソフトウェア開発サービスの評価

開発サービスプロバイダによって提供されるソフトウェア開発サービスの評価を行う。開発サービスプロバイダは、リソース量、リソース状態など、リアルタイムに更新される自身のリソース情報を保持する。リソース情報に基づき、サービスの実行可能性や品質を評価する。

3) ソフトウェア開発サービスの決定

ソフトウェア開発サービスの評価に基づいて利用するサービスを決定する。サービスコンシューマはサービスの品質とプロジェクト要求を比較し、プロジェクトの要求を満たすサービスを決定する。複数の開発サービスプロバイダに対してサービスコンシューマがネゴシエーションを行い、サービス実行のスケジュールを調整する必要がある。調整による変更が生じる度に、サービスの評価を行う。一連のサービス決定のプロセスは非常に複雑になる。

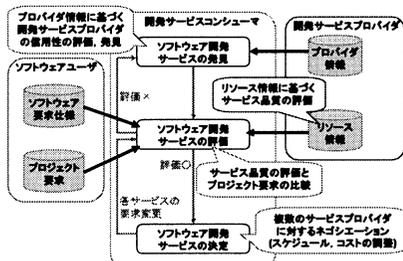


図9 プロジェクトプロセスモデルの設計

(3) 実行可能モデルへの変換

定義したプロジェクトプロセスモデルをプロセス実行エンジンで実行可能なモデルに変換する。これにより、マシンでソフトウェア開発プロジェクトを実行、管理することが可能となる。

(4) プロジェクトの実行とモニタリング

プロセス実行エンジンでプロジェクトを実行する。実行中はプロセス実行モニタによって実行をモニタリングする。プロジェクト実行中に KPI の数値から異常を検知したら、サービスの組み換え、変更によって対応する。また、プロジェクト実行後のモニタリングデータから、ソフトウェア開発プロセスの改善点を分析して、ソフトウェア開発の雛形を改善する。

5.6. ソフトウェア開発ビジネスメタモデル

ビジネス全体の枠組みを表すビジネスメタモデル [12]に基づいて、ソフトウェア開発ビジネスメタモデルを定義する(図 10)。ビジネスメタモデルをソフトウェア開発の要素として特殊化した。

ソフトウェア開発メタモデルはソフトウェア開発全体に含まれる要素とその関連を表している。ソフトウェア開発において定義されるモデルは、ソフトウェア開発ビジネスメタモデルの要素と関連を満たす必要がある。以下に、ソフトウェア開発ビジネスメタモデルの要素について説明する。

- (1) 環境: ソフトウェア開発の動機付けになる社会的、技術的な要因。
- (2) 目的: ソフトウェア開発プロジェクトにおける目的。
- (3) 責務: ソフトウェア開発プロジェクトにおける職務間の契約。
- (4) 成果: ソフトウェア開発プロジェクト中の成果物。
- (5) 職務: ソフトウェア開発プロジェクトの実行に必要な役割。
- (6) プロセス: ソフトウェア開発プロセスなどのソフトウェア開発の作業順序。
- (7) 資源: ソフトウェア開発に必要なコスト、開発チームのメンバなどのリソース。
- (8) 機能: 設計や実装などの、ソフトウェア開発に必要な工程。
- (9) 組織: ソフトウェア開発を行う組織。

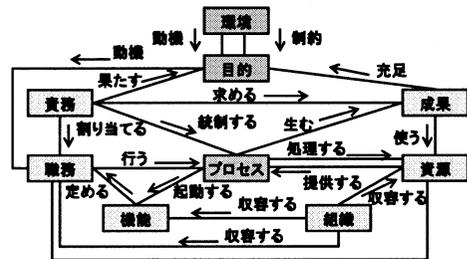


図10 ソフトウェア開発ビジネスメタモデル

6. BPMN の意味拡張

6.1. ソフトウェア開発に対する BPMN の意味拡張

ビジネスを特殊化したモデルとして、ソフトウェア開発を捉え、BPM のフレームワークに基づく、ソフトウェア開発の実行と管理を提案した。そこで、BPM フレームワークの出発点であるビジネスプロセスモデルとして、ソフトウェア開発プロセスをモデル化可能と考える。BPMN はビジネスを視覚的に表現するための表記法であり、実行可能言語である BPEL へのマッピングが定義されている[7,11]。BPMN にソフトウェア開発に対する意味拡張を行い、ソフトウェア開発プロセスのモデル化を行う。

(1) フローオブジェクトの意味拡張

ソフトウェア開発プロセスにおけるフローオブジェクトは、ソフトウェア開発の作業を定義するための図形要素である(表 2)。また、アクティビティには表 3 に示す属性が与えられており、アクティビティのタイプ、実行者、入力能力などが定義可能である。

表 2 フローオブジェクトの意味拡張

要素	表記法	説明(BPMN)	説明(ソフトウェア開発)
イベント	○	プロセスの進行中に発生する事象を表す。	ソフトウェア開発中に発生する事象を表す。イベントはソフトウェア開発の実行を表す。
アクティビティ	□	企業で実行される作業を表す。	ソフトウェア開発で実行される作業を表す。
ゲートウェイ	◇	フローの分岐や収束を制御する。	分岐や収束によって、ソフトウェア開発の実行を制御する。

表 3 アクティビティの属性の例

属性	説明
ActivityType (Task/SubProcess)	ActivityType属性はTaskかSub-Processを設定する。
Task:String	
Performers (0-n):String	1つかそれ以上のPerformersが設定される。Performersの属性はアクティビティを実行するリソースかアクティビティに責任を持つリソースを定義する。
InputSets (0-n):InputSet	InputSets属性は、アクティビティに入力されるデータ集合を表す。0個以上のInputSetsを定義できる。各InputSetが満たされるとアクティビティの実行が許可される。
OutputSets (0-n):OutputSet	OutputSets 属性は、アクティビティから出力されるデータ集合を定義する。0個以上のOutputSetsを定義できる。アクティビティの完了時に、OutputSetsのうちの1個だけが生成される。どのOutputSetを生成するかは判断は、アクティビティの実績によって異なる。

(2) 接続オブジェクトの意味拡張

ソフトウェア開発プロセスにおける接続オブジェクトは、ソフトウェア開発の作業の流れや相互作用を表す(表 4)。シーケンスフローは、組織内の作業の実行順序を定義する。シーケンスフローで接続された一連のアクティビティがプロセスである。メッセージフローは、組織間のメッセージによる相互作用を定義する。個々のプロセスはシーケンスフローでは独立しているが、メッセージフローによって接続される。

(3) スイムレーンの意味拡張

スイムレーンはモデリング要素を分類する図形要素である。プールとスイムレーンを組み合わせることでソフトウェア開発の作業を実行する組織や役割、リソースなどを定義する(表 5)。

(4) 成果物の意味拡張

成果物はソフトウェア開発プロセスに関する補足情報を提供する図形要素である(表 6)。主にデータオブ

ジェクトによって、ソフトウェア開発の作業に関連する入出力を定義する。

表 4 接続オブジェクトの意味拡張

要素	表記法	説明(BPMN)	説明(ソフトウェア開発)
シーケンスフロー	→	アクティビティが実行される順序を示す。	ソフトウェア開発の作業の実行順序を示す。 制約条件: 1つのソースとターゲットをもち、フローオブジェクトを接続する。プールの境界を超えることができない。サブプロセス内と外のオブジェクトを接続できない。
メッセージフロー	○...▷	関係者間のメッセージの送受信を表す。	組織間のメッセージを示す(組織間のインタラクション)。 制約条件: 異なるプール内のオブジェクト、異なるプールの境界、異なるプールのサブプロセス内のオブジェクト、異なるプールのサブプロセスの境界に接続可能。同じプールの境界内に存在するオブジェクトを接続できない。
関連	⇒	情報をフローオブジェクトに関連付ける。	情報をフローオブジェクトに関連付ける。ソフトウェア開発の作業の入出力を示す。

表 5 スイムレーンの意味拡張

要素	表記法	説明(BPMN)	説明(ソフトウェア開発)
プール	□	プロセスの関係者を表す。	ソフトウェア開発プロセスの関係者を表す。
レーン	□	プール内の分割区画を表す。	ソフトウェア開発プロセスの関係者の分割区画を表す。

表 6 成果物の意味拡張

要素	表記法	説明(BPMN)	説明(ソフトウェア開発)
データオブジェクト	□	アクティビティが要求、生産するものに関する情報を提供する。	ソフトウェア開発の作業が要求、生産するものに関する情報を提供する。
グループ	□	アクティビティのグループ分けを表す。	ソフトウェア開発の作業のグループ分けを表す。
注釈	□	モデルが読み手に補足情報の提供を表す。	モデルが読み手に補足情報の提供を表す。

6.2. 階層的なソフトウェア開発プロセスの記述

ソフトウェア開発を実行する組織は階層的な構造であるため、ソフトウェア開発プロセスもこれに対応して階層的に記述する必要がある(図 11)。

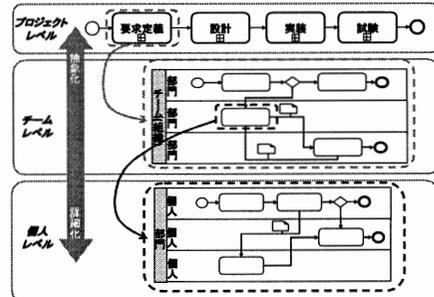


図 11 階層的なソフトウェア開発プロセスの記述

BPMN ではサブプロセスによって、ソフトウェア開発の作業の実行順序を階層的に記述できる。また、スイムレーンを用いて、プロセスの実行者をプロジェクトレベルから、個人レベルまで定義することが可能である。以上から、ソフトウェア開発プロセスの階層的な記述が可能である。

7. 例題の記述と評価

7.1. 例題の概要

開発プロセスモデル化の例題[6]を、BPMN を用いて記述した。まず、例題で定義されている作業とアクティビティを直接対応付け、記述した(図 12)。

例題は、ソフトウェア変更プロセスに注目し、ソフトウ

ウェアシステムのデザインの変更、コードの変更、単体試験に焦点を絞っている。一部のデータ交換は共有ファイルを用いて行われる。プロセス開始時には、監査機関(CCB)によって要求変更が承認されている。プロジェクトマネージャ(PM)が変更のスケジューリングとタスクの割り当てを行うことで、開発エンジニア(DE)と品質保証エンジニア(QAE)がソフトウェア変更プロセスを開始し、コードが単体テストを合格することで終了する。

7.2. ソフトウェア開発のサービス化に基づくプロセス

図 12 に示す例題の開発プロセスを、ソフトウェア開発のサービス化の観点からモジュール化を行い、図 13 に記述する。共通フレームで定義されているアクティビティをモジュール化の基準とする。

デザインの変更とデザインレビューを、デザインの変更アクティビティとして統合する。同様に、コード変更と単体試験パッケージの変更、単体試験を、コード変更と単体試験アクティビティとして統合する。

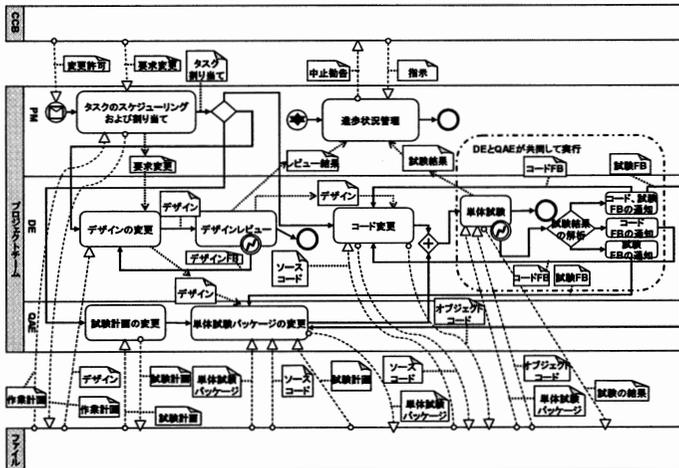


図 12 BPMN による例題の記述

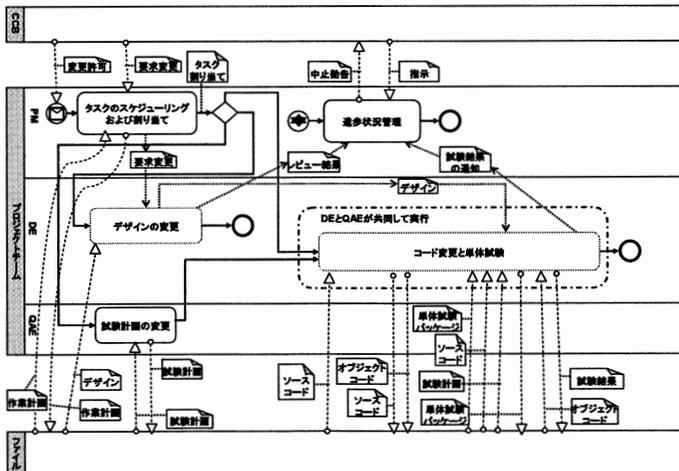


図 13 BPMN によるソフトウェア開発のサービス化に基づく記述

7.3. サービス化の評価

例題の開発プロセス(図 12)と、ソフトウェア開発サービスに基づく開発プロセス(図 13)を比較し、制御、成果物の観点からサービス化の評価を行う。

(1) 制御の観点

例題の開発プロセスでは、単体試験の実行に関して、ゲートウェイによる制御の情報が記述される。また、デザインの変更とデザインレビューの二つのアクティビティ間で繰り返し構造が存在する。さらに、単体試験後に実行する試験結果の解析から、単体試験パッケージの変更とコード変更へのフィードバックが発生し、複数のアクティビティにまたがった繰り返し構造が存在する。ソフトウェア開発のサービス化に基づく開発プロセスでは、これらの制御構造の詳細や繰り返し構造がアクティビティ内部に隠蔽される。そのため、各アクティビティは内部で実行が完結する、自律した独立性の高いアクティビティとなる。しかし、開発プロジェクトのきめ細かいスケジュールを実行するためには、図 12 に示すような、

サービス単位よりも詳細な開発プロセスのモデルが必要である。

(2) 成果物の観点

図 12 では、デザインの変更とデザインレビューのアクティビティ間で、レビューされていないデザインやデザインフィードバックのデータが交換される。図 13 のデザインの変更は、要求変更と変更前のデザインを入力として、変更後、レビューで承認されたデザインを出力する。サービス化に基づいて開発プロセスを記述することで、アクティビティ内部にこれらの中間成果物が隠蔽される。各アクティビティは、それ自身が完結した、意味を持つ成果物を出力する機能を提供することが可能になる。

8. 評価

8.1. SOA に基づくソフトウェア開発の妥当性評価

例題の記述と評価から、開発工程をサービス単位として考えることで、ソフトウェア開発サービスは、自律した独立性の高い、意味を持つ機能を提供することができる。分散化する複数の組織に、ソフトウェア開発サービスとして作業を割り当てることで、組織内で作業の実行が完結する。また、データ交換の詳細もサービス内部に隠蔽される。ソフトウェア開発サー

ビスの協調として、ソフトウェア開発プロジェクトの実行と管理を行うことで、各サービス内部の詳細な作業やデータ交換が隠蔽され、サービス単位の制御とサービス間のインタラクションに集中することができる。

今後、ソフトウェア開発サービスのインタフェースが、標準のインタフェース記述言語によって定義されることでソフトウェア開発サービスは疎結合となり、サービスの協調による統一的な開発プロジェクトの実行と管理が可能となる。

8.2. モデル化プロセスの妥当性評価

提案したモデル化プロセスでは、ソフトウェア開発プロセスの設計を、プロジェクト独立モデルとプロジェクト固有モデルに分離した。プロジェクト独立モデルは、ソフトウェア開発サービスを組み合わせて設計する。例題の記述と評価から、開発プロセスをサービスレベルに抽象化することで、詳細の情報が隠蔽され、開発プロジェクト全体の管理が容易になる。

しかし、ソフトウェアサービスと異なり、ソフトウェア開発サービスは人的リソースにより実行されるため、サービスのリソース状態によってサービスの実行可能性や品質が変化する。そこで、開発プロジェクトの実行をするプロジェクト固有モデルでは、各開発サービスプロバイダのリソース情報に基づいたサービス品質の評価による、ソフトウェア開発プロセスの設計を提案している。これらによって、サービスレベルの開発プロジェクトの管理と、サービスのリソース状態を考慮した開発プロジェクトの実行が可能となる。

8.3. メタモデルに基づく開発プロセスの評価

BPMN によるソフトウェア開発プロセスはソフトウェア開発プロセスの要素とその関連を表すソフトウェア開発ビジネスメタモデルを満たす必要がある。BPMN によるソフトウェア開発プロセスの記述では、スイムレーンによってプロセスの実行者である職務や割り当てられるリソース、それらを収容する組織の記述が可能である。また、データオブジェクトによってプロセスから生成される成果物を記述できる。ソフトウェア開発を実行する機能であるソフトウェア開発サービスは、BPMN によるソフトウェア開発プロセスに基づいて起動される。このように、ソフトウェア開発ビジネスメタモデルによって、モデルが表す要素と関連の確認が可能である。

9. 今後の課題

提案したモデル化プロセスでは、プロジェクト独立モデルとプロジェクト固有モデルの分離を提案したが、詳細の設計方法については定義していない。プロジェクト独立モデルに基づくプロジェクト固有モデルの設計を体系化する必要がある。また、ソフトウェア開発サービスのインタフェースモデルを検討する必要がある。

10. まとめ

本稿では、ソフトウェア開発をサービスの複合体と考え、ソフトウェア開発におけるサービス指向の適用を提

案した。従来のビジネスにおける SOA と BPM のモデルと対応する、ソフトウェア開発におけるモデルを提案することにより、ソフトウェア開発における SOA と BPM の適用を明確にした。さらに、対応するモデルに基づいて、ソフトウェア開発のモデル化から実行、管理を実現するサービス指向ソフトウェア開発モデル化プロセスを提案した。また、意味拡張した BPMN を用いた、ソフトウェア開発プロセスの例題の記述と分析を通して、提案するモデル化方法論の妥当性を示した。

参考文献

- [1] M. Aoyama, A Business-Driven Web Service Creation Methodology, Proc. IEEE SAINT 2002, IEEE Computer Science, Feb. 2002, pp. 225-228.
- [2] 青山 幹雄, サービス開発のサービス化を実現する統一サービスシステム USS (Unified Service Systems) の提案, ウィンターワークショップ 2009・イン・宮崎 論文集, Jan. 2009, pp. 53-54.
- [3] T. Erl, Service-Oriented Architecture, Prentice Hall, 2005.
- [4] D. F. Ferguson, et al., Enterprise Business Process Management - Architecture, Technology and Standards, BPM 2006, LNCS Vol. 4102, Springer, pp. 1-15.
- [5] J. D. Herbsleb, et al., Global Software Development, IEEE Software, Vol. 18, No. 2, Mar. - Apr. 2001, pp. 16-20.
- [6] M. I. Kellner et al., Software Process Modeling Example Problem, Proc. of 6th ISPW, Oct. 1990, IEEE Computer Science, 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].
- [7] M. B. Juric, et al., Business Process Driven SOA using BPMN and BPEL, Packet Publishing, 2008
- [8] 田岡 賢輔ほか, BAM・可視化経営の実践, 日経 BP 企画, 2006.
- [9] M. Kloppmann, et al., WS-BPEL Extension for People BPEL4People, Jul. 2005, <http://www.ibm.com/developerworks/>.
- [10] C. Legner, et al., Transforming Inter-Organizational Business Processes into Service-Oriented Architecture Method and Application in the Automotive Industry, Proc. KiVS 2007, Feb. 2007.
- [11] 加藤 正人, BPMN によるビジネスプロセスモデリング入門, ソフト・リサーチ・センター, 2006.
- [12] D. W. McDavid, A Standard for Business Architecture Description, IBM Systems Journal, Vol. 38, No. 1, 1999, pp. 12-31.
- [13] L. Osterweil, Software Processes are Software too, Proc. ICSE 1987, IEEE Computer Science, Mar. 1987, pp. 2-13.
- [14] SEC, 共通フレーム 2007, オーム社, 2007.