

状況に応じた CASE ツール選択の支援に関する研究

2010SE233 竹内優俊 2010SE272 山本佳宏

指導教員：沢田篤史

1 はじめに

Service Oriented Architecture(以下, SOA)[1] はサービスを一つの部品としてとらえ, その部品を組み合わせたシステムを作るための開発方法論である. SOA に基づくシステムの開発において, 生産性, 信頼性の高い製品の構築にミドルウェアの利用は必要不可欠である. 本研究室では SOA に基づくシステムのためのアプリケーションプラットフォームのプロダクトライン化 [6] に関する研究が行なわれている. Product Line Software Engineering(以下, PLSE)[3] はソフトウェア開発の生産性や製品の信頼性を向上するための方法論である. 石川らの研究 [5] では, ソフトウェア開発のツール統合による PLSE 環境の提案をしており, ソフトウェア開発の生産性や信頼性を支援している.

現在 SOA に基づくシステムのためのアプリケーションプラットフォームのプロダクトラインの仕様は, アプリケーション開発プロセスをコア資産としていない. 開発者の経験によって開発プロセスを構築するので, 開発工程が円滑に移行できないという問題がある. 異なるミドルウェアを扱う場合, アプリケーション開発における開発プロセスも変化する. 開発者はアプリケーションプラットフォームのミドルウェアが変わる度に, 適切な開発プロセスの組み換えや開発ツールの選択をする必要がある. 開発プロセスの構築やツール選択は開発者の経験に依存し, 誤った開発プロセスやツール選択は手戻りなどによる開発期間の延長, 製品の信頼性の低下につながる.

本研究の目的は, 開発プロセスの構築によるアプリケーションプラットフォームのプロダクトライン化の支援である. 可変性に対応した開発プロセス構築の枠組みを提案する.

本研究では, ミドルウェアの選択と開発プロセスの依存関係, ミドルウェア間の依存関係の整理を行なった. ミドルウェアを組み合わせてアプリケーションプラットフォームを形成することで, アプリケーション開発の具体的なプロセスが決定する. 図 1 に開発プロセスの決定の流れを示す.

ミドルウェアの選択によって開発プロセスが変化することに着目し, 選択したミドルウェアに対応した開発プロセスが構成可能であると考えた. ミドルウェアの依存関係を整理することにより, 適切なツール選択が可能になると考えた. 状況に応じて適切なツールを使用することで, 製品の信頼性の向上や開発工程の手戻りなどを防ぐ. 今回, PLSE を支援するサービスを本研究室で提案している SOA に基づくアプリケーションプラットフォームを事例に用いた.

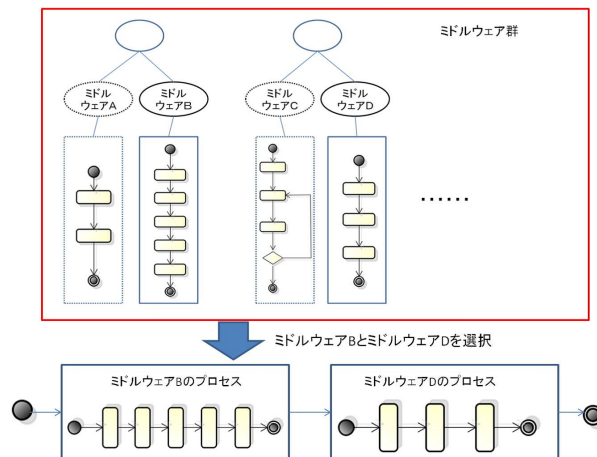


図 1 ミドルウェアの選択によるアプリケーション開発プロセス

2 背景技術

本章では, 背景技術である CASE ツール, PLSE についてと, BPMN について説明する.

2.1 CASE ツール

Computer Aided Software Engineering tool(以下, CASE ツール)とはコンピュータを利用したソフトウェア開発を支援するツールの総称である. CASE ツールを導入することで一部の作業を自動化できたり, 論理的な矛盾や設計の完全性などをチェックすることができる. CASE ツールは適応する開発工程や範囲で上流 CASE ツール, 下流 CASE ツール, 統合 CASE ツールに分類される.

2.2 PLSE

PLSE とは, 再利用可能な部品をコア資産として利用し, ソフトウェアを系統的に開発する開発方法論である. コア資産とは, 過去の開発によって蓄積された, 再利用可能な知識や部品のことである. コア資産の再利用により, 開発の生産性と製品の信頼性の向上が期待できる. PLSE はドメインエンジニアリング・アプリケーションエンジニアリング・管理により構成される.

- ドメインエンジニアリング
製品系列を分析し, 共通部分と可変部分を定義し再利用可能なコア資産を開発する.
- アプリケーションエンジニアリング
製品開発に求められる要求を基にコア資産から再利用可能な部分を選択し, 製品の開発を行なう.
- 管理
製品系列の範囲の特定とコア資産の保守を行なう.

2.3 BPMN

Business Process Modering Notation(以下, BPMN)[2]とは, ビジネスプロセスを表記する記法であり, 一般に業務フローやシステムフローを表記するさいに使用される. BPMN を用いた具体的な例を図2に示す. 図2はある商品の受注を受けてから代金の支払い方法を現金かクレジットか確認し, 商品を引渡すまでの業務フローを示している.

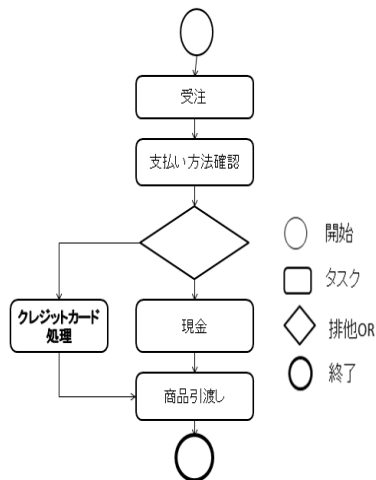


図2 BPMN で表現した支払い業務フローの例

BPMN は実行言語である Business Process Execution Language(以下, BPEL) の生成が可能である. BPEL は, 定義されたプロセスに従って制御オーケストレーションを記述する標準実行言語である. BPMN では BPEL へのマッピング・ルールが規定されており BPMN から BPEL を生成するツールなどが開発されている. 業務側で発生したプロセスの変更を迅速かつ確実に ITシステムへ実装することが可能となる.

3 ミドルウェア選択プロセス

本研究では, ミドルウェア選択プロセスを BPMN を用いてモデル化した. 本章で BPMN を利用した理由, プロダクトフォーチャモデル, ミドルウェアの選択プロセスのモデル化について説明する.

3.1 BPMN を使用した理由

BPMN を使用した理由は, 以下である.

- プロセスの複雑なセマンティクスをテクニカルユーザーとビジネスユーザーの両方が理解しやすい表記法
- BPMN によるビジネスプロセスモデルを駆動して BPEL などを生成することが可能
- メッセージによるアクティビティの駆動を記述できるので, 各プロセスを独立な記述が容易
- プロジェクトごとに標準化やパターン化を行なうことで, ビジネスプロセスモデルの品質向上と均質化を図ることが可能

3.2 プロダクトフィーチャモデル

SOA システムに基づくシステムのためのアプリケーションプラットフォームのプロダクトラインでは, 共通性や可変性を仕様モデルとして記述している. このアプリケーションプラットフォームのプロダクトラインにおける仕様モデルを図3に示す. この仕様モデルでは Operating Environment Layer 層にミドルウェアの可変性を記述する. アプリケーションプラットフォームのプロダクトラインは, ミドルウェアの差異を吸収し特定の技術にとらわれないシームレスな開発を目的としている.

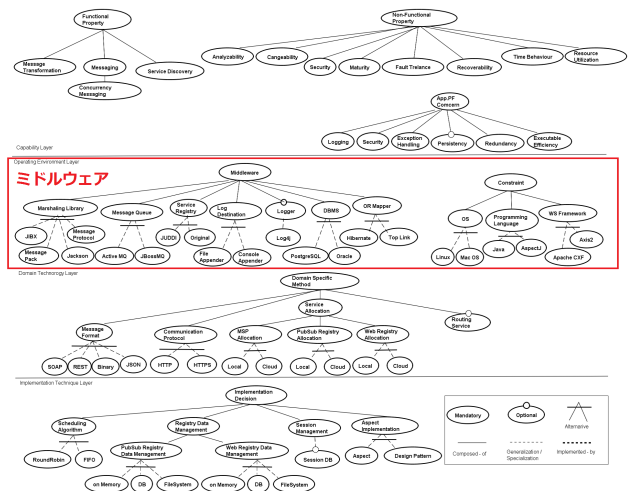


図3 アプリケーションプラットフォームのプロダクトラインにおける仕様モデル

3.3 ミドルウェア選択プロセスのモデル化

プロダクトフィーチャモデルを作成するプロセスを考察することで, ミドルウェア選択プロセスが構築できると考えた. 本研究では BPMN でミドルウェアの選択プロセスを, モデルの利用目的に応じて3つのレベルでモデル化した. 3つのレベルに分割してプロセスを表現する理由として, BPMN は1つのビジネスプロセスを視覚的に分かりやすいフロー図から BPEL として実行可能なレベルにまで詳細に描写した図まで描写することができる. そのギャップを埋める目的で我々は3つのレベルを用いてプロセスを描写する. 各レベルで描写すべき事柄を定義し, 実際に BPMN を用いてハイレベル, ミドルレベル, ローレベルでモデル化を行なった.

3.3.1 ハイレベルビジネスプロセスモデル

プロセスの可視化を目的とする. 各プロセスの成果物間の依存関係から順序性を分析する. ハイレベルでモデル化した図を図4に示す.

3.3.2 ミドルレベルビジネスプロセスモデル

ハイレベルで記述したプロセスの細分化および分析, 改善を目的とする. 手戻りが発生しうるプロセスと, 再考す

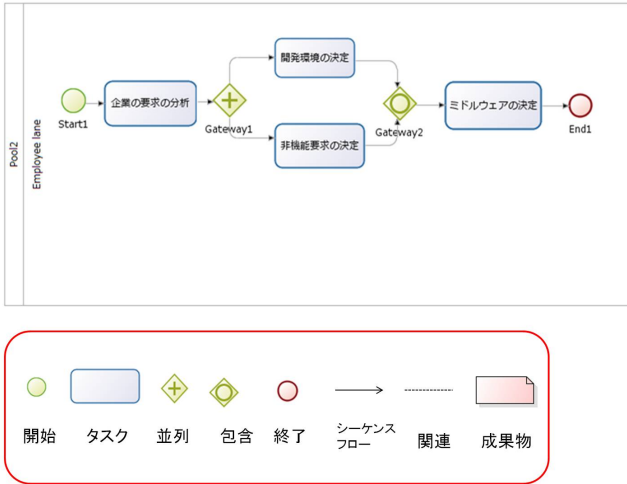


図4 ハイレベルで記述されたミドルウェア選択プロセス

べきプロセスの分析を行なう。ミドルレベルでモデル化した図を図5に示す。

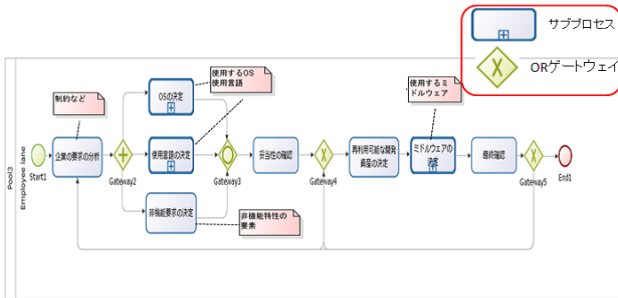


図5 ミドルレベルで記述されたミドルウェア選択プロセス

3.3.3 ローレベルビジネスプロセスモデル

BPELを生成する目的で記述する。システム的な振る舞いを詳細に記述し、BPELの生成に必要な情報を付加する。ローレベルでモデル化した図を図6に示す。

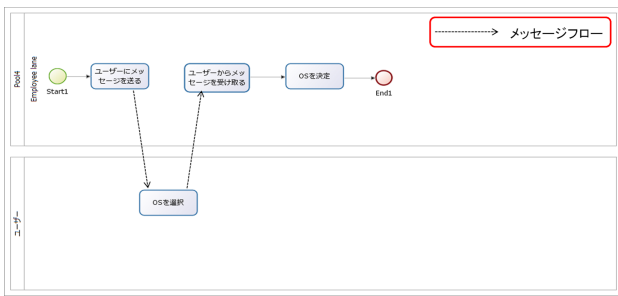


図6 ローレベルで記述されたミドルウェア選択プロセスのサブプロセス

4 ミドルウェアを選択したさいのアプリケーション開発プロセス

ミドルウェアを選択したさいに開発プロセスが反自動的に決定することに着目し、可変性点に対応した開発プロセス自動構築の枠組みを提案する。

4.1 ミドルウェアに付随するプロセスの差異

ミドルウェアの選択によって開発プロセスが異なることに着目し、選択したミドルウェアに基づき開発プロセスが構成可能であると考えた。図7ではミドルウェアを選択したさいに現れる開発プロセスの差異を示す。

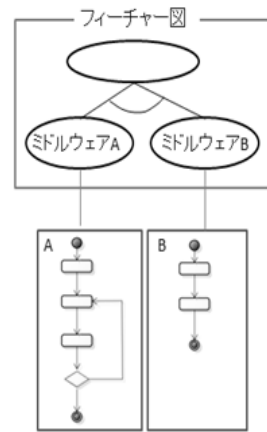


図7 ミドルウェアを選択したさいに現れる開発プロセスの差異

4.2 ミドルウェア間の依存関係

特定のミドルウェア間では成果物の入出力関係が現れる。この成果物の入出力関係を整理することにより、開発プロセスにおけるミドルウェア実行の優先順位や並列実行の可否が決定できると考えた。ミドルウェアの依存関係として先行研究 [4] を参考に以下の3パターンを我々は定義した。定義した依存関係を図8に示す。図中の太い矢印はミドルウェア A とミドルウェア B の成果物の作成開始タイミングと終了タイミングを表している。入力関係では A の成果物作成が終了した次点で B の成果物が作成開始できる。同期関係では A の成果物の独立した部分 a ができた段階で B の成果物の作成を開始し、相互に情報を交換しつつ成果物の作成を行なう。入力と同期の合成関係では A の成果物の独立した部分 a ができた段階で B の成果物の作成を開始できるが、A の成果物が完成しないと B の成果物も完成しない。また、成果物の関係からプロセスの手戻り先が予測できると考えた。図9に考案した手戻りの予測パターンを示す。図9で矢印の始点は成果物の手戻りが必要になるタイミングを示し、終点は手戻り先を示す。例として入力関係では B で手戻り画必要になったさいに手戻りする先の候補として自身の成果物の開始時点、もしくは A の成果物に不具合があると考え、A の成果物の作成

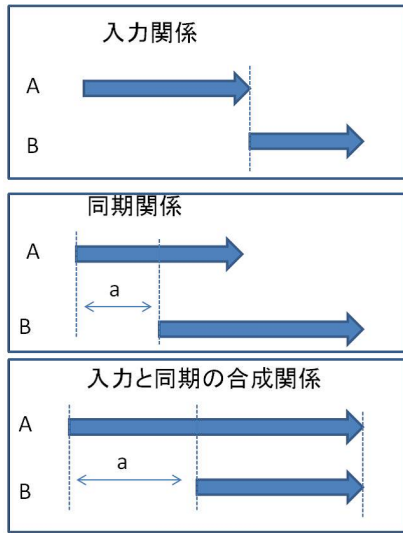


図8 成果物間の依存関係

開始時点へ手戻りすると考えた。また、Aの成果物で手戻りが必要になったさいはAの成果物の作成開始時点へ手戻りすると考える。

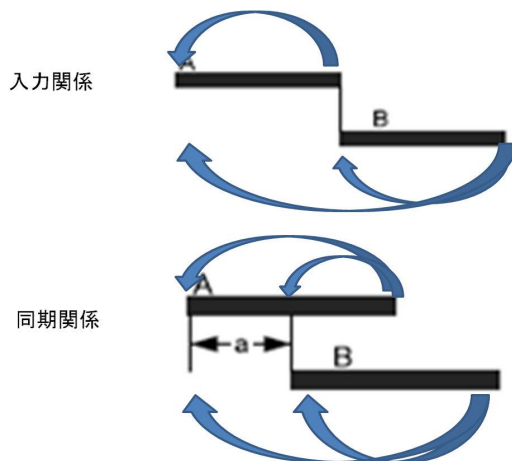


図9 手戻りのパターン

4.3 アプリケーション開発プロセス

選択したミドルウェアを依存関係を元に組み合わせ、ミドルウェア間の優先順位が確保された各ミドルウェアに付随したプロセスを対応させる事でアプリケーション開発プロセスが構築できると考えた。

5 考察

ミドルウェアを選択することで具体的な開発プロセスが構築されたと考えた。ミドルウェアの選択と開発プロセスの依存関係、ミドルウェア間の依存関係の整理を行なった。ミドルウェアの依存関係から開発プロセスにおけるミドルウェアの順序性の確保やプロセスの手戻り予測が可能であると確認できた。今後の課題として、我々の提案した枠組みにより構築された開発プロセスに対応した、適切な

開発支援ツールの選択が挙げられる。ツールと開発プロセスは互いに影響し合い、不適切なツール選択は不要な工程を生んだり、製品に悪影響を及ぼす。適切なツール選択を支援する枠組みは石川らの研究 [5] をはじめとし、多々あり、それらを我々の提案した枠組みに取り入れることで適切なツール選択を保障できるのではないかと考えられる。それに加え、本研究では BPMN を用いてプロセスの記述を行なっている。ツール選択の枠組みを開発プロセスの中にツール選択プロセスとして組み込み、BPMN から生成した BPEL を用いてツール選択の自動化が実現できると考える。開発プロセスで見られる手戻りについては本研究では成果物の作成終了時点のみに注目しており、対して実際の現場ではあらゆるタイミングで手戻りが確認でき、これだけでは十分であるとは言えない。手戻りの予測について、実際の事例を基にして考察し直せば更に多くの手戻りについて予測が可能になるのではないかと考えられる。また、ミドルウェアの並列実行性について、本研究は並列処理の可否までしか考慮されておらず、作業効率性などを考慮した並列実行処理の枠組みは提案できていない。なので複数人での開発のさい、担当するプロセスの割り振りを制御する枠組みを提案する余地がある。

6 おわりに

本研究ではミドルウェアの選択によって開発プロセスが変化することに着目し、選択したミドルウェアに基づいた開発プロセス構築の枠組みを提案した。そのためにミドルウェアの選択と開発プロセスの依存関係、ミドルウェア間の依存関係を整理した。

参考文献

- [1] D. Krafzig, D. Slama, and K. Banke, *Enterprise SOA*, Prentice Hall, 2004.
- [2] H. Micheal, *Essential Business Process Modeling*, O'Reilly Japan, 2006.
- [3] K. Pohl, G. Bockle, and F. J. Linden, *Software Product Line Engineering: Foundations, Principles, And Techniques*, Springer-Verlag, 2005.
- [4] 飯田元, 田中康, 松本健一, “成果物間の関連に着目した開発プロセスモデル: PReP” 情報理工学会論文誌. vol. 46 no. 5 pp. 1233-1245
- [5] 石川拓也, 山内達也, “ツール統合による PLSE 環境の提案,” 南山大学情報理工学部ソフトウェア工学科, 2013.
- [6] 江坂篤侍, 野呂昌満, 沢田篤史, “SOA に基づくシステムのためのアプリケーションプラットフォームのプロダクトライン化に関する研究,” 情報処理学会研究報告. ソフトウェア工学研究会報告, vol. 2013-SE-179, no. 25, pp. 1-6, 2013.