

# アスペクト指向に基づく SOA の考察

## — サービス指向アプリケーション開発のためのアーキテクチャの提案 —

2008MI025 花井 萌

2008MI129 松本 真子

指導教員 野呂 昌満

### 1 はじめに

一般にエンタープライズ系システムに要求される最も重要な非機能特性として、ビジネス環境の変化に対応する柔軟性が挙げられる。柔軟性を向上させるためには、システムを矛盾なくモジュール分割する必要がある。サービス指向アーキテクチャ(以下、SOA)が、ビジネス環境の変化に対応して柔軟なシステムのモジュール化の技術として注目されている。SOA に基づくシステムのアーキテクチャの研究が行なわれている。代表的な研究として Papazoglou らの Service-Oriented Design and Development Methodology[1] が挙げられる。Papazoglou らの研究では、サービス指向開発方法論を Web サービス開発ライフサイクル階層として定義した階層アーキテクチャ(以下、LA)を提案している。

Papazoglou らの LA を基にアプリケーション構造を定義すると、層を跨いで横断する関心事が存在することがある。Papazoglou らの LA では、システムの構造を利用者視点から手続きとしてとらえたサービスを、開発者視点からとらえたコンポーネントを用いて実装する。オブジェクト指向技術の利点から、一般的にシステムはオブジェクト指向設計すべきであるとされている。本研究では、コンポーネントをオブジェクト指向技術でモジュール分割したオブジェクトとして考える。SOA に基づくシステムには異なる関心事が存在するので、横断する関心事が存在する。横断する関心事が存在することで、柔軟な変更が困難となる。ビジネス環境の変化に対応してアプリケーションを柔軟に変更するためには、アプリケーションを矛盾なくモジュール分割する必要がある。

本研究の目的は、サービス指向アプリケーション開発の支援である。Papazoglou らの LA を基にアプリケーションアーキテクチャ(以下、APA)を提案する。本研究では、非機能要求をプラットフォームで実現していることを前提とし、アプリケーション設計に注目する。実装技術・実行環境に依存しないアプリケーションのソフトウェア部品・部品間の関係を APA として定義する。APA と APA を設計するための開発プロセスを提案することで、より生産性を向上させることができると考える。

本研究は、APA とアプリケーション設計プロセスを提案する。サービスとオブジェクトモデルのそれぞれの再利用性を向上させるために、アスペクト指向技術を用いて Papazoglou らの LA を改良する。アスペクト間記述と同等なオブジェクトコンテナ(以下、OC)を定義

し、サービスとオブジェクトに横断する関心事を分離する。サービスとオブジェクト間に OC を設け、サービスの代わりに OC がオブジェクトに対してメッセージを送るとする。OC を仲介することで、サービスとオブジェクトそれぞれのメッセージ形式に変換することが可能となり、横断的関心事が分離できる。改良した LA を基にアプリケーションの構成要素を明確にし、APA を定義する。APA を基にサービス設計とオブジェクト設計を並行して行なうアプリケーション設計プロセスを定義する。事例検証を行ない、APA とアプリケーション設計プロセスについて考察する。

本研究の結果として、アスペクト指向技術を用いることによってサービスとオブジェクトを分離した APA を定義できた。APA とアプリケーション設計プロセスを提案することで、柔軟に変更可能なアプリケーション開発の支援を行なうことができたと考える。

### 2 関連研究

SOA に基づくシステムの開発方法に関する研究について説明する。代表的な研究として Papazoglou らの研究 [1] を取り上げる。サービス開発ライフサイクル階層として提案されている LA を図 1 に示す。Papazoglou らの研究は、サービス開発ライフサイクル階層を用いてサービス指向開発方法論を提案している。

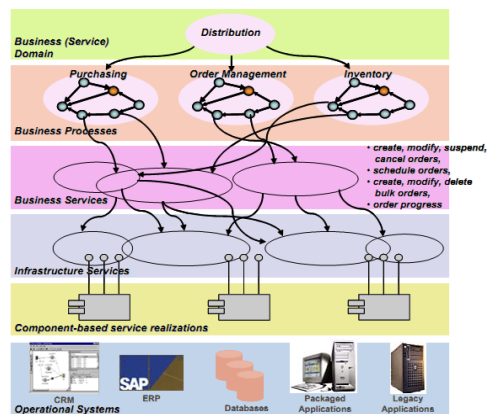


図 1 Papazoglou らの研究で提案されている Web サービス開発ライフサイクル階層

### 3 アーキテクチャの提案

#### 3.1 階層アーキテクチャの改良

Papazoglou らの LA を改良し、LA の再定義を行なう。アスペクト指向技術を用いてサービスとオブジェクト

トの分離を行ない、LA の再定義を行なう。

アスペクト指向技術を用いてサービスとオブジェクトを分離する。分離する方法を図 2 に示す。アスペクト間記述と同等な OC を仲介し、OC がサービス・オブジェクトそれぞれのメッセージ形式に変換する。OC を仲介することにより、モジュール化技術の異なるメッセージ通信に関する記述がサービスとオブジェクトから分離される。すなわち、サービスとオブジェクトを分離することが可能となる。

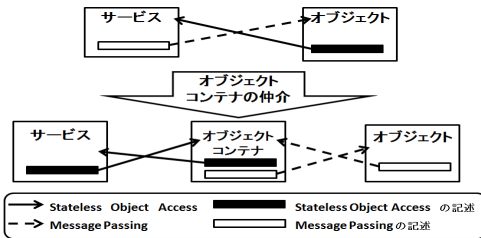


図 2 OC を用いた横断的関心事の分離

Papazoglou らの LA を改良し、新たに LA を定義する。定義した LA を図 3 に示す。変更した点は、2 点で

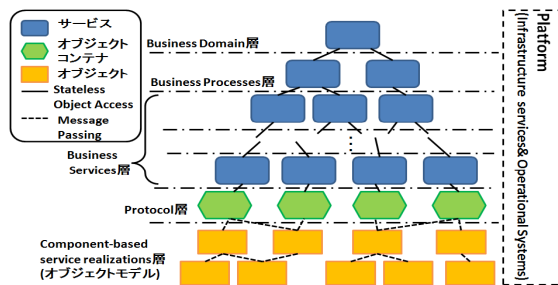


図 3 提案する LA

ある。1 つ目は、OC 層の設けたことである。2 つ目は、Infrastructure Services 層と Operational Systems 層をまとめてプラットフォームとして縦に設けことである。本研究では、非機能要求をプラットフォームで実現していることを前提とする。非機能要求をプラットフォームにおいて実現するので、アプリケーションではビジネスロジックの実現について焦点を当てる。実装技術・実行環境に依存しないアプリケーション構造を定義する。

図 3 より、サービス・OC・オブジェクトが LA におけるアプリケーションの構成要素となる。

### 3.2 アプリケーションアーキテクチャの提案

改良した LA において必要となるソフトウェア部品・部品間関係を定義し、APA を提案する。

改良した LA を基に、アプリケーションの構成要素と構成要素の関係を定義する。サービスをユーザと相互作用するサービスと相互作用しないサービスに分類することができる。ユーザと相互作用するサービスを Web アプリケーションとし、相互作用しないサービスを Web

サービスとする。まとめると、以下の 4 点がアプリケーションの構成要素となる。

- Web アプリケーション
- Web サービス
- OC
- オブジェクト

LA を基にアプリケーション構成要素の関係を定義した。提案する APA を図 4 に示す。

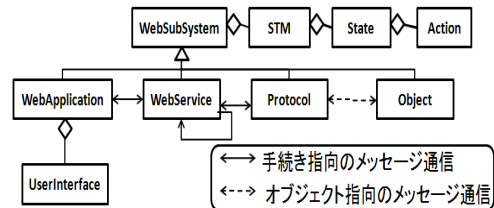


図 4 提案する APA

Web サービス・Web アプリケーション・OC を状態遷移機械 (以下、STM) で実現する。STM で実現することによって、サービスの状態とメッセージに関する記述を状態とアクションとして別々の部品とすることが可能となる。別々の部品とすることで、メッセージを送るモジュールを変更する時に変更箇所を特定しやすくなる。

## 4 アプリケーション設計プロセスの提案

提案した APA からアプリケーションを設計するプロセスを、Papazoglou らの研究で提案されている開発方法論を基に提案する。柔軟に変更可能なアプリケーションを開発するためには上流工程が重要となることから、本研究ではアプリケーションの分析・設計プロセスを提案する。提案する分析・設計フェーズと各フェーズの入出力を図 5 に示す。

分析フェーズでは、システムの全体像の明確化を目的に、ビジネスモデリングとビジネスモデリングの結果を基にしたソフトウェア化の範囲の分析を行なう。分析フェーズは、2 つのサブフェーズで構成される。ビジネスプロセス分析フェーズでは、要求に対する業務全体のプロセスとデータフローを明確にするために、アクティビティ図とデータフロー図を作成する。システム分析フェーズでは、ビジネスモデルからソフトウェア化する範囲を明確にするためにユースケース図とユースケース記述を作成し、サービスを抽出する。

設計フェーズでは、一般的なオブジェクト指向設計にビジネスの設計を並行に行なう。ビジネスの視点からは、サービスを手続き指向設計する。サービス設計とオブジェクト設計を並行して行ない、OC で構造変換することでサービスとオブジェクトモデルを分離できる。全てのサービスと OC は、STM で実現する。本研究では、組込みシステムのためのアスペクト指向ソフトウェアアーキテクチャスタイル (以下、E-AoSAS++) を提

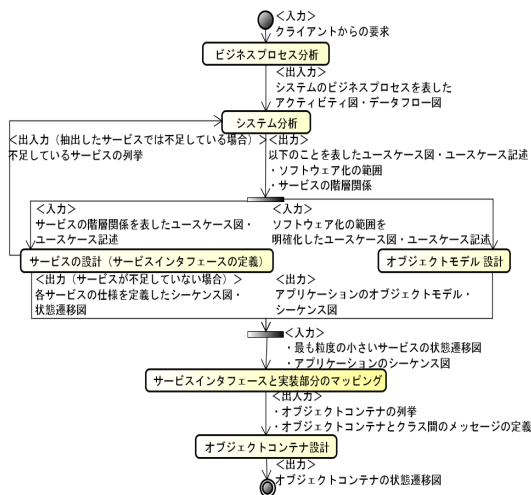


図5 提案する分析・設計フェーズ

案している。E-AoSAS++[2]では、STMを設計するために、シーケンス図を作成し、シーケンス図を基に状態遷移図を作成する。サービス設計フェーズとOC設計フェーズでは、E-AoSAS++の記法に従い、サービスとOCのシーケンス図と状態遷移図を作成する。設計フェーズは、4つのサブフェーズで構成される。サービス設計フェーズでは、システム分析で抽出したサービスのSTMを設計する。オブジェクトモデル設計フェーズでは、サービスの実装部分をオブジェクト指向設計し、クラス図とシーケンス図を作成する。サービス設計フェーズとオブジェクトモデル設計フェーズは並行して行なう。サービスインタフェースと実装部分のマッピングフェーズでは、サービスとオブジェクトモデルを対応付け、OCを定義する。コンテナ設計フェーズでは、定義したOCのSTMを設計する。

### 5 事例検証: 出庫受付システム

提案したアプリケーション設計プロセスに沿ってシステムの分析・設計を行なうことで、提案したアプリケーション設計プロセスが妥当であるかを確かめる。事例には、図6に概要を示した酒類販売会社の出庫受付システムの受付側の仕事を行なうシステム(以下、出庫受付システム)を用いる。

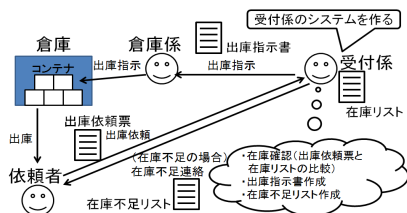


図6 出庫受付システムの概要

出庫受付システムのLAを図7に示す。出庫受付シス

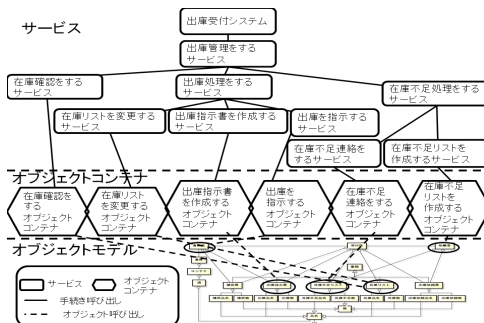


図7 出庫受付システムのLA

テムのAPAを図8に示す。

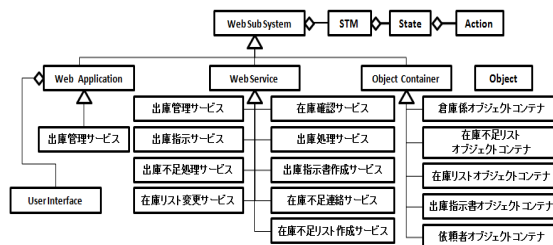


図8 出庫受付システムのAPA

提案したアプリケーション設計プロセスに従って設計を行なうことで、サービスとオブジェクトが分離したAPAを設計することができた。従って、提案したアプリケーション設計プロセスは妥当と考える。

### 6 考察

本研究で提案したLAとAPAの妥当性を確認する。本研究では、容易に変更可能なアプリケーションの構造を定義するためにLAとAPAを提案した。以下の2点に着目してLAとAPAの妥当性を確認する。

- OCの導入によるサービスとオブジェクトの再利用性
- サービスをSTMで表現することによるサービスの再利用性

#### 6.1 OCの導入によるサービスとオブジェクトの再利用性

OCを導入することによるサービスとオブジェクトの再利用性の向上について考察する。本研究は、サービスとオブジェクト間の横断的関心事を分離するために、アスペクト間記述であるOCを導入した。サービスとオブジェクトのメッセージ通信をOCで仲介することによって、メッセージの形式を変換する。OCを仲介することでサービスとオブジェクトを分離し、サービスとオブジェクトのモジュール化を矛盾なく行なうことが可能となる。最も粒度の小さいサービスと1:1で対応するOC

を定義することで、サービスとオブジェクトが互いの変更の影響を受けない構造を定義することができた。サービスとオブジェクトが互いに変更の影響を受けないことから、サービスとオブジェクトの再利用性が向上すると考える。

OCの導入によるサービスの再利用性の向上について例を用いて説明する。電車の時刻検索システムを構成するサービスを再利用して新たにシステムを構築する際のサービスの再利用性について考える。電車の時刻検索システムを構成するサービスのひとつである「乗車する電車の発着時刻を特定する」サービスの再利用性について考える。OCを導入しない場合のサービスの再利用性について考える。OCを導入しない場合の「乗車する電車の発着時刻を特定する」サービスとオブジェクトの動的挙動を表したシーケンス図を図9に示す。サービス

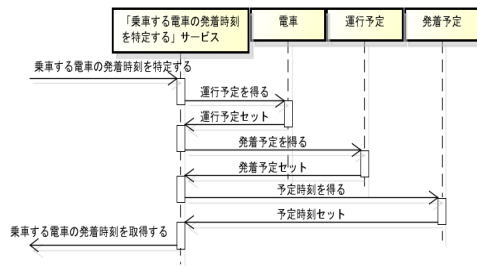


図9 OCを導入しない場合の「乗車する電車の発着時刻を特定する」サービスのシーケンス図

とオブジェクトが直接メッセージ通信を行なうことで、サービスとオブジェクトの依存関係が強くなる。サービスはオブジェクトモデルに依存することから、オブジェクトモデルが変更した場合にサービスは変更の影響を受ける。最も粒度の小さいサービスと1:1で対応するOCを導入する場合のサービスの再利用性について考える。OCを導入する場合の「乗車する電車の発着時刻を特定する」サービスとオブジェクトの動的挙動を表したシーケンス図を図10に示す。OCを導入することによって、

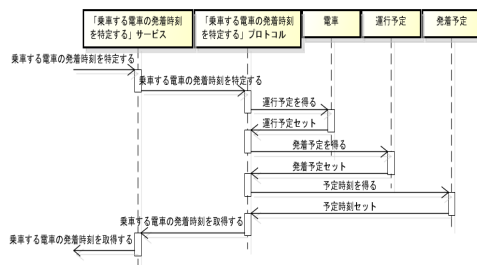


図10 OCを導入する場合の「乗車する電車の発着時刻を特定する」サービスのシーケンス図

オブジェクトに対するメッセージ送信の記述をサービスに記述する必要がなくなる。従って、サービスはオブ

ジェクトモデルの変更の影響を受けない。

OCの導入によるオブジェクトの再利用性の向上について例を用いて説明する。電車の時刻検索システムのオブジェクトモデルを再利用して新たにシステムを構築する際のオブジェクトの再利用性について考える。OCを導入することによって、サービスに対するメッセージ送信の記述をオブジェクトに記述する必要がなくなる。従って、サービスの変更によりサービスを構成するオブジェクトの組合せを変更する可能性はあるが、オブジェクト自体やオブジェクト間の関係は影響を受けない。

OCを導入することによってサービスとオブジェクトは互いの変更の影響を受けないことが確認できた。結果として、OCを導入することでOCを導入しないよりもサービスとオブジェクトの再利用性が向上することが確認できた。従って、提案したLAとAPAは妥当であると考えられる。

## 6.2 サービスをSTMで表現することによるサービスの再利用性

サービスをSTMで実現することによるサービスの再利用性について考察する。サービスをSTMで実現することで、サービスからのメッセージとメッセージの送信先をアクションとして部品化することが可能となる。サービスを再利用して新たにシステムを構築する場合について考える。メッセージ通信に関する記述が部品としてまとめられていることから、メッセージを送信するサービスに変更があった場合でもアクションのみが変更を必要とする。サービスの状態については、再利用可能となる。サービスをSTMで実現することで、ビジネス環境の変化に対応してアプリケーションの構成変更を容易に行なえると考えられる。従って、提案したLAとAPAは妥当であると考えられる。

## 7 おわりに

本研究はPapazoglouらのLAを改良し、サービスとオブジェクトを分離したAPAを提案した。APAにおいて、SOAに基づくシステムに必要なソフトウェア部品・部品間の関係を定義することで、詳細な開発プロセスを定義することができた。APAと開発プロセスを提案することによって、サービス指向アプリケーション開発の生産性が向上することができたと考える。今後の課題は、ステートレスを実現することとコードの自動生成について考察することである。

## 参考文献

- [1] M.P. Papazoglou, and W. Heuvel, "Service-Oriented Design and Development Methodology," *IJWET*, vol. 2, no. 4, pp. 1-17, 2006.
- [2] 加藤大地, 蜂巣吉成, 沢田篤史, 野呂昌満, "アスペクト指向に基づくソフトウェアアーキテクチャの文書化方式," 知能ソフトウェア工学研究会 (KBSE), vol.108, no.449, pp.55-60, 2009.