

# SOAに基づくシステムのためのマルチプラットフォーム対応のアーキテクチャの妥当性確認

M2012MM004 花井萌

指導教員：野呂昌満

## 1 はじめに

本研究室では、SOAに基づくシステム（以下、SOAシステム）のためのアプリケーションプラットフォーム（以下、App.PF）のプラットフォーム化に関する研究がおこなわれている。伴ら [2] は、江坂ら [1] によって定義された App.PF のプラットフォームにおいて、プラットフォームアーキテクチャ（以下、PLA）の自動生成を目指している。仕様上のコンポーネント群とアーキテクチャ上のコンポーネント群の対応関係を整理することで、仕様モデルとアーキテクチャ間の相互の追跡性を確保している。

SOA システムの開発において、定義された App.PF のプラットフォームの一般性が確認されていない。普遍的な SOA システムの開発をおこなうために、プラットフォームが有用である保証がされていない。そこで、特定のミドルウェアを利用した App.PF の構築にプラットフォームが有用である事を確認する必要がある。プラットフォームを利用した製品の構築を繰り返し、必要に応じて結果をプラットフォームにフィードバックすることでプラットフォームを洗練する必要がある。

本研究の目的は、App.PF の PLA の妥当性確認である。プラットフォームの SOA システムの開発における有用性確認の一部として PLA の妥当性確認をおこなう。プラットフォームソフトウェアエンジニアリング（以下、PLSE）は、アーキテクチャを中心としてソフトウェアを開発するための工学である。したがって、PLA の妥当性を確認する事でプラットフォームの有用性の保証が可能であると考えられる。

PLA に基づいて事例システムに対する要求を実現する App.PF を構築する事で、PLA の妥当性確認をおこなう。事例として ATM 監視システム、歯科診療業務サポートシステムを用いる。PLA に基づいて、プラットフォームアーキテクチャ（以下、PA）を構築する。PA を構築する際に仕様モデル上のコンポーネント群の選択の組み合わせによって PA のコンポーネント群が一意に決定することを確認する。PA 上のコンポーネント群が一意に決定することを確認することで、仕様モデル上のコンポーネント群とアーキテクチャ上のコンポーネント群の対応関係の正しさを確認する。また、プラットフォームで定義された Variation Point と Variant が十分であることを確認する。結果として、PLA に記述されている、仕様上のコンポーネント群の選択の組み合わせに応じて決定する各 Aspect の変動コンポーネントの定義の正しさを保証できると考える。さらに、構築した PA に基づいた設計が可能であることを確認することで、PLA に記述されている Aspect 間の関係の正しさを保証できると考える。プラットフォーム構築の過程において、必要に応じてプラットフォームを洗

練する。

本研究の成果として、2つの事例において PLA が妥当であることを確認できた。さらに、新たに Variant の存在を確認でき、製品の構築結果をフィードバックしたことで App.PF のプラットフォームを洗練することができた。

## 2 App.PF のプラットフォーム化に関する研究

SOA システムの開発は、生産性、信頼性を確保するためにミドルウェアを利用しておこなうことが一般的である。SOA システムを実現するための枠組みは、既存のミドルウェア製品によって提供されている。SOA システムの開発者は、システムの開発環境や企業の開発戦略、政策などの制約から特定のミドルウェア製品を選択する。各ミドルウェア製品は、前提とするアーキテクチャや実現する非機能要求が異なる。したがって、ミドルウェア製品の選択によってシステムの開発形態が制約されるという問題が生じる。

江坂らは、App.PF においてプラットフォームごとの差異を吸収することで、特定の技術や製品から独立した普遍的な SOA システムの開発を可能とする事を目指している。さらに、App.PF をプラットフォーム化することで、開発に利用するプラットフォームを柔軟に変更可能とする事を目指している。App.PF の PLSE のプロセスを図 1 に示す。江坂らは、App.PF のプラットフォーム

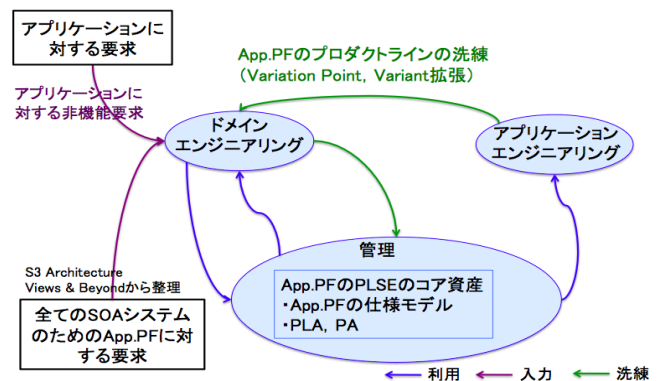


図 1 App.PF の PLSE のプロセス

んのコア資産として、次の 3 つを定義している。

### ● PLA

App.PF の Concern によって定義されるコンポーネント（以下、Aspect）間の関係と Aspect における Variation Point, Variant の選択の組み合わせに応じて変動する Aspect のコンポーネント群を記述

- PA  
 プロダクト特有の機能・非機能要求を実現するための Aspect とデータ構造の定義を記述
- 仕様モデル  
 App.PF に対する関心事とその変動性の分析結果を記述

伴らは PLA の自動生成を目的に、仕様モデルとアーキテクチャの相互の追跡性を確保することを目指している。プロダクトラインにおける共通性、可変性を再整理し、仕様上の Variation Point (変動点) と Variant (変動点の実現候補となる値) を仕様モデルに記述している。要求に応じたアーキテクチャの構築を可能とするために、仕様上のコンポーネント群とアーキテクチャ(プロダクトモデル)上のコンポーネント群の対応関係を整理している。仕様モデル上のコンポーネント群と対応するアーキテクチャ上のコンポーネント群は多対多の関係にある。

### 3 PLA の妥当性確認方法

本研究では、2つの事例を用いて、App.PF のプロダクトラインのコア資産である PLA からプロダクトが構築可能なことを確認する。PLA の妥当性確認のためのプロセスを図 2 に示す。まず、事例に用いたシステムに対す

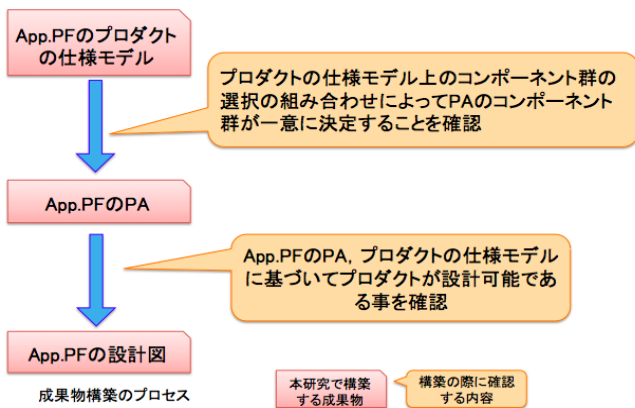


図 2 PLA の妥当性確認のためのプロセス

る要求(非機能要求, 開発環境)を入力に仕様モデルのフィーチャを枝刈りし、プロダクトの仕様モデルを作成する。次に、プロダクトの仕様モデルと PLA に基づいて事例に用いたシステムに対する要求を実現する App.PF の PA を構築する。PA を構築する過程で、仕様モデル上の Variant の選択の組み合わせによって PA のコンポーネント群が一意に決定することを確認する。PA のコンポーネント群が一意に決定することを確認することで、次の 2つを示すことができると考える。

- 仕様モデルのコンポーネント群とアーキテクチャ上のコンポーネント群の対応関係が正しい
- プロダクトラインで定義された Variation Point と Variant が十分

結果、PLA に記述されている、仕様上のコンポーネント群の選択の組み合わせに応じて決定する各 Aspect の変動コンポーネントの定義の正しさを保証できると考える。

次に、構築した PA とプロダクトの仕様モデルに基づいて事例の仕様を満たした App.PF の設計をおこなう。PA とプロダクトの仕様モデルに基づいてプロダクトの設計が可能であることを確認することで、PLA に記述されている Aspect 間の関係の正しさを保証できると考える。結果として、2つの事例において PLA が妥当であるといえる

と考える。事例に用いたシステムのための App.PF 構築の過程で、必要に応じてプロダクトラインを洗練する。新たに Variation Point の存在を確認した場合は、Variation Point と非機能要求の実現関係を整理する。Variation Point の実現候補となる Variant を調査し、仕様モデルを洗練する。調査結果から仕様上の Variant とアーキテクチャ上の Variant となるコンポーネント群の対応関係を整理し、仕様モデルとアーキテクチャ間の相互の追跡性を確保する。新たに Variant の存在を確認した場合は、仕様モデルを洗練し、App.PF のプロダクトラインで定義されている枠組みに従って新しく部品を追加する。

### 4 事例を用いた PLA の妥当性確認

本研究では、事例として ATM 監視システムと歯科診療業務サポートシステムを用いる。

#### 4.1 ATM 監視システムを用いた PLA の妥当性確認

ATM 監視システムは、銀行の ATM 端末の挙動監視、障害対応を目的としたシステムである。常時稼働の ATM を監視するシステムであり、オペレータが直接操作することから、重視する非機能要求は、時間効率性、信頼性(成熟性、障害許容)とする。App.PF に対する主な制約として、OS, ミドルウェア, 言語が挙げられる。OS は、Mac OS を利用する。Web Service Framework には、Axis2 を利用する。言語は、Java を利用する。

プロダクトの仕様モデルと PLA に基づいて、ATM 監視システムのための App.PF の PA を構築した。図 3 に ATM 監視システムのための App.PF の PA を示す。構

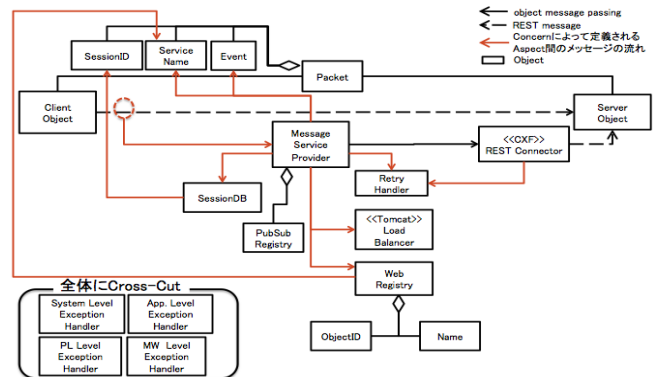


図 3 ATM 監視システムのための App.PF の PA

築した ATM 監視システムのための App.PF の PA とプロダクトの仕様モデルに基づいて設計をおこなった。図 4 に ATM 監視システムのための App.PF の設計図を示す。

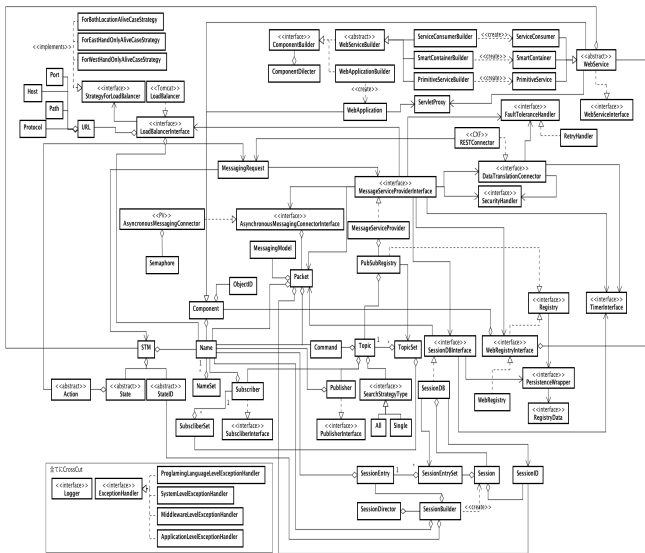


図 4 ATM 監視システムのための App.PF の設計図

#### 4.2 歯科診療業務サポートシステムを用いた PLA の妥当性確認

歯科診療業務サポートシステムは、歯科診療所のスタッフがおこなう業務を支援することを目的としたシステムである。医療情報システムであることから、重視する非機能要求は、国が定めるガイドラインに従い、セキュリティ、信頼性（成熟性、障害許容、回復性）、保守性（解析性）、時間効率性とする。App.PF に対する主な制約条件として、OS、ミドルウェア、言語が挙げられる。OS は、Windows を利用する。言語は、C# を利用する。開発環境として .NET Framework を利用し、Web Service Framework には WCF を利用する。

プロダクトの仕様モデルと PLA に基づいて、歯科診療業務サポートシステムのための App.PF の PA を構築した。図 5 に歯科診療業務サポートシステムのための App.PF の PA を示す。構築した歯科診療業務サポートシステム

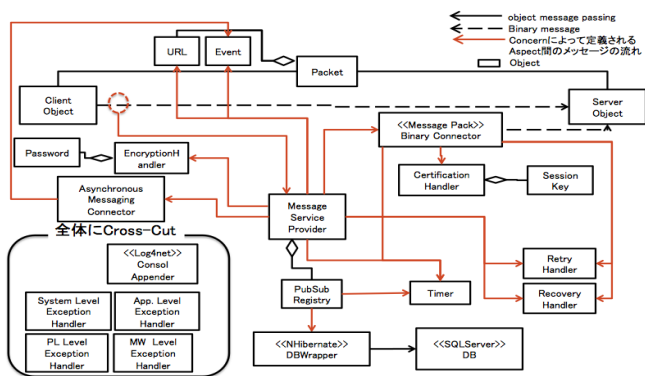


図 5 歯科診療業務サポートシステムのための App.PF の PA

のための App.PF の PA とプロダクトの仕様モデルに基づいて設計をおこなった。図 6 に歯科診療業務サポートシステムのための App.PF の設計図を示す。

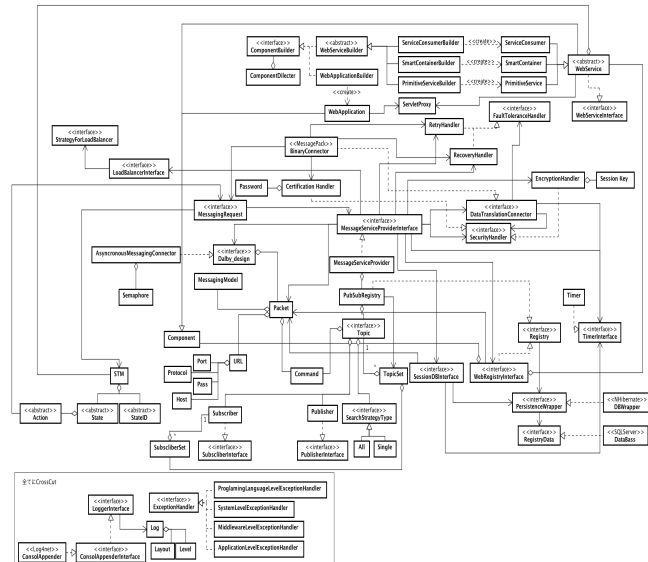


図 6 歯科診療業務サポートシステムのための App.PF の設計図

### 5 考察

事例システムのための App.PF の PA 構築の結果、プロダクトの仕様モデルと PLA に基づいて PA のコンポーネント群が一意に決定づくことを確認した。本誌では仕様上のコンポーネント群と PA 上のコンポーネント群の対応関係の一部分を図 7 に示す。他の仕様上のコンポーネント群と PA 上のコンポーネント群の対応関係については、本稿に記載する。事例に用いた 2 つのシステムの

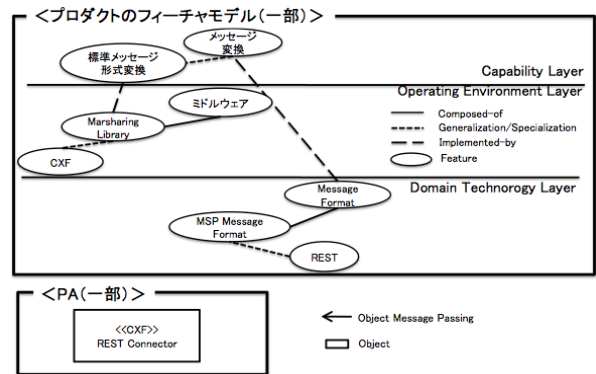


図 7 仕様上のコンポーネント群と PA 上のコンポーネント群の対応関係（一部分）

ための App.PF の開発において、次の 2 点を保証する事ができた。

- 仕様モデルのコンポーネント群とアーキテクチャ上のコンポーネント群の対応関係が正しい
- プロダクトラインで定義された Variation Point と Variant が十分

結果、PLA に記述されている、仕様上のコンポーネント群の選択の組み合わせに応じて決定する各 Aspect の変動コンポーネントの定義の正しさを保証できたと考える。



事例システムのための App.PF の設計の結果，構築した PA とプロダクトの仕様モデルに基づいて App.PF を設計する事が可能であることを確認した．図 8 に，PLA から設計レベルの静的構造を定義した過程（一部）を示す．図 8 で挙げた Concurrency Aspect 以外の Aspect においても同様に，構築した PA とプロダクトの仕様モデルに基づいて App.PF を設計する事が可能であることを確認できた．他の Aspect の設計結果については，本稿に記載する．結果として，PLA に記述されている Aspect 間の関係の正しさを保証できたと考える．

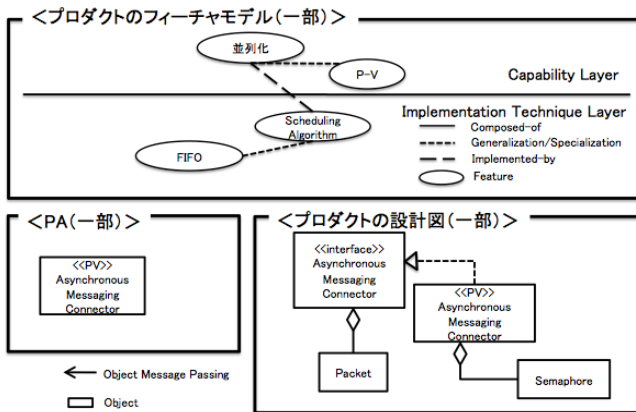


図 8 PLA から設計レベルのプロダクトの静的構造定義の過程（Concurrency Aspect）

歯科診療業務サポートシステムのための App.PF 構築の過程で，新たに仕様上の Variant の存在を確認した．新たに存在を確認した仕様上の Variant を表 1 に整理する．仕様上の Component 群と PA 上のコンポーネント群

表 1 新たに存在を確認した仕様上の Variant

フィーチャ	サブフィーチャ
OS	Windows
Programming Language	C#
Runtime System	CLR
Web Service Framework	WCF
HTTP Server	.NET Framework Class Library
Logger	Log4net
OR Mapper	NHibernate
DBMS	SQL Server

の対応関係の枠組みに従い，新たに存在を確認した仕様上の Variant に対応する Aspect の静的構造を定義した．図 9 に，新たに存在を確認した仕様上の Variant である Log4net フィーチャに対応する Consol Appender Aspect の静的構造を示す．プロダクトの構築結果をフィードバックすることで App.PF のプロダクトラインを洗練することができた．

事例検証の結果，ATM 監視システム，歯科診療業務サポートシステムにおいて App.PF のプロダクトラインが妥当である事が確認できた．本研究で事例として利用した 2 つのシステムは，いずれもエンタープライズ系システムである．エンタープライズ系システムは，企業の業

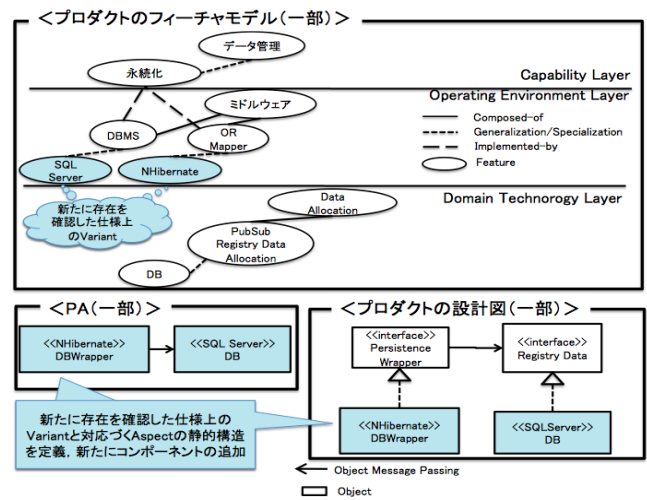


図 9 新たに存在を確認した仕様上の Variant (Log4net) に対応する Consol Appender Aspect の静的構造

務システムや情報システムなど，大規模かつ社会基盤を支える情報システムである．エンタープライズ系システムには，高い信頼性が求められる．本研究で事例として利用した 2 つのシステムにおいても，信頼性を重視しており，それに伴ってミドルウェアや実現技術の選択を行なった．したがって，事例検証の結果からエンタープライズ系システムにおいて PLA が妥当であるといえると考ええる．

## 6 おわりに

本研究は，江坂らの研究で定義された App.PF のプロダクトラインの有用性確認の一部として PLA の妥当性確認をおこなった．事例検証の結果，エンタープライズ系システムにおいて App.PF の PLA が妥当であることを確認した．さらに，新たに Variant の存在を確認でき，プロダクトの構築結果をフィードバックしたことで App.PF のプロダクトラインを洗練することができた．今後の課題は，他の事例を用いた事例検証とプロダクト構築結果をフィードバックすることによるプロダクトラインの洗練を繰り返すことであると位置づけている．SOA に関連する技術は，今後拡大する事が予想される．将来にわたりマルチプラットフォーム対応の App.PF を提供し続ける為に，プラットフォームの進化に伴いプロダクトラインを進化させる事で保守し続ける必要がある．

## 参考文献

- [1] 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”，2013 年情報処理学会，2013．
- [2] 伴昌樹，吉川翔平，“SOA 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—アスペクト指向に基づくプロダクトアーキテクチャの自動生成—”，南山大学大学情報理工学部，2013 年度卒業論文，2014．