

歯科レセプト処理システムの開発 —SOA に基づくシステムのアーキテクチャ設計支援—

M2010MM042 玉木佑一

指導教員：野呂昌満

1 はじめに

Service-Oriented Architecture(以下, SOA) はビジネス環境の変化に対して柔軟に変更可能なシステムの開発技術である。Georgakopoulos らは, SOA システムのリファレンスアーキテクチャを示している [1]。一般に非機能要求等も考慮して, リファレンスアーキテクチャからシステムアーキテクチャを設計することは容易ではない [8]。アーキテクチャが存在する事を前提とする Product Line Software Engineering(以下, PLSE)[6] においても, ドメイン分析においてアーキテクチャ設計が必要な場合は少なくないと我々は認識している。仕様とアーキテクチャとの追跡性を確保しておくことで, アーキテクチャ設計を支援することができる。

本研究の目的は, 非機能要求を満たすシステムアーキテクチャ設計のためのガイドラインを定義することである。SOA に基づくシステムにおける非機能要求と, 非機能要求の実現に必要なアーキテクチャのコンポーネントを整理する。PLSE の考えに基づき, 非機能要求を考慮した仕様モデルと, プロダクトラインアーキテクチャを定義する。定義した仕様モデルとプロダクトラインアーキテクチャとの追跡性を確保することで, それらをプロダクトアーキテクチャ設計のためのガイドラインとして定義する。これにより, 非機能要求を考慮したアーキテクチャのコンポーネントの配置を追跡することができる。

仕様モデルとアーキテクチャとで, 扱う関心事や表すコンポーネントの粒度が異なることから, 追跡性を確保することは一般に困難である。互いの本質を損なうことなく, 対応付け可能な仕様モデルとアーキテクチャを定義することが, 追跡性確保の鍵となる。アーキテクチャ設計にアスペクト指向技術を適用することで解決を図る。アスペクト指向技術を用いることで, 支配的分割に横断する仕様モデルで扱う関心事をアスペクトとして分離することができ, 追跡性の確保が容易になる。我々は, FORM[5]で提案されているフィーチャモデルを, 広く一般に用いられているとの認識から, 仕様モデル記述に用いる。アーキテクチャの記法には Oldevik が提案するアスペクト指向アーキテクチャの記法 [4] を用いる。

事例として, OJL(On the Job Learning) として開発に取り組んでいる歯科レセプト処理システムを取り上げる。歯科レセプト処理システムとは, 歯科診療所における診療報酬請求業務を支援するシステムである。定義したガイドラインを利用し, SOA に基づく歯科レセプト処理システムのアーキテクチャを設計する。

2 歯科レセプト処理システム概要

歯科レセプト処理システムとは, 歯科診療所の歯科診療報酬請求事務をサポートするシステムである。歯科レ

セプト提出業務の軽減と迅速化を目的として, レセプト電算処理システムが提案されている。我々は歯科レセプト提出業務に必要な診療事務処理(レセプトチェック, 患者負担医療費精算, レセプト作成)を支援するツールを Web サービスとして開発してきた。歯科レセプト処理システムは, 社会保険診療報酬支払基金が公布する診療報酬請求の標準仕様 [9] に準拠している。

3 ガイドラインの定義

3.1 ガイドラインの概要

非機能要求を考慮した仕様モデルとプロダクトラインアーキテクチャを定義し, 追跡性を確保することでそれらをプロダクトアーキテクチャ設計のためのガイドラインとして定義する。図1にガイドラインの概要を示す。シ

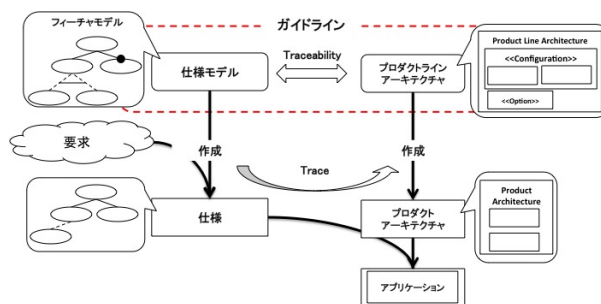


図1 ガイドラインの概要

ステム開発の流れとして, 要求に応じて仕様モデルから仕様を作成し, 仕様を基にプロダクトアーキテクチャを設計する。仕様とプロダクトアーキテクチャを基に, アプリケーションを作成する。

仕様モデルとプロダクトラインアーキテクチャを定義し, 対応関係を明確にし, 追跡性を確保しておく。これにより, 仕様モデルから仕様を作成する際に, 必要となるアーキテクチャのコンポーネントを追跡することが可能となり, プロダクトアーキテクチャ設計が支援できる。

本研究では, 仕様モデルの非機能要求を表すコンポーネントと, 非機能要求の実現に必要なプロダクトラインアーキテクチャのコンポーネントを対応付け, 追跡性を確保する。SOA システムに求められる位置透過性やスケラビリティ等の, システムアーキテクチャでの実現が求められる非機能要求を対象とする。

3.2 アスペクト指向技術の適用

仕様モデルとアーキテクチャとの追跡性の確保を目的として, アーキテクチャ設計に対しアスペクト指向技術を適用する。仕様モデルとアーキテクチャとで扱う関心事を揃えることで, 対応付けが容易になり, 追跡性を確

保することができる。関心事を揃える方法として、仕様モデルで扱う関心事を考慮したアーキテクチャ設計を行なうことが挙げられる。仕様モデルで扱う関心事がアーキテクチャの支配的分割に対し横断することが、この方法の問題となる。この問題に対し、アーキテクチャ設計にアスペクト指向技術を適用することで解決を図る。システムの支配的分割から、仕様モデルで扱う関心事をアスペクトとして分離し定義することで、仕様モデルとの対応付けが容易なアーキテクチャを設計する。

SOA システムのアーキテクチャにおいて、非機能要求は横断的関心事となりうることから、非機能要求を考慮したモジュール化、および追跡性の確保は困難である。プロダクトラインアーキテクチャ設計にアスペクト指向技術を適用し、非機能要求毎に実現に必要となるアーキテクチャのコンポーネント群をアスペクトとして定義する。これにより、図2に示すように、非機能要求を表す仕様モデルのコンポーネントとアスペクトを1対1で対応付けることができ、追跡性の確保が容易になる。

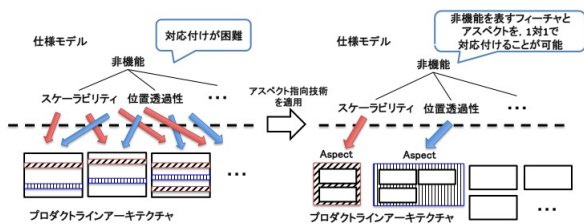


図2 アスペクト指向技術の適用による追跡性の確保

3.3 非機能要求を考慮した仕様モデルの定義

SOA に基づくシステムに求められる非機能要求の整理を行なう。SOA に基づくシステムに求められる特性を実現するためのシステムアーキテクチャに関する研究が行なわれている [1, 3, 7]。提案されているシステムアーキテクチャから、SOA に基づくシステムに求められる非機能要求と、実現に必要なアーキテクチャのコンポーネント、実現に用いる技術を整理した。本研究では次に示す非機能要求を取り扱う。

- 位置透過性 (Location Transparency)
- スケーラビリティ (Scalability)
- 並行性 (Concurrency)
- 拡張性 (Expandability)
- セキュリティ (Security)
- 耐故障性 (Fault Tolerance)
- 信頼性 (Reliability)
- 相互運用性 (Interoperability)
- 正確性 (Accuracy)

整理した結果を基に、フィーチャモデルを作成した。非機能要求を Capability Layer に配置する。非機能要求の実現に用いる技術を Implementation Technique Layer に配置し、非機能要求と関係付ける。図3にフィーチャモデルを示す。

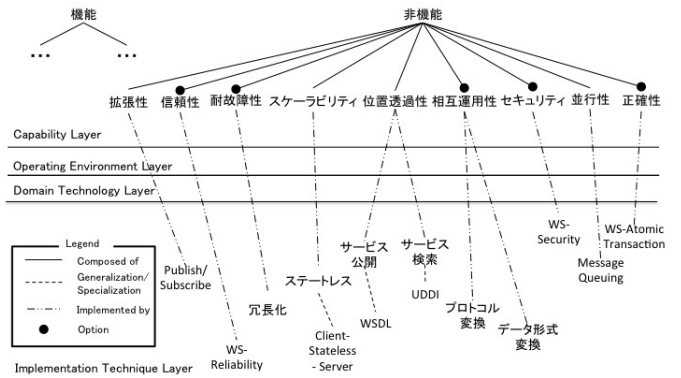


図3 非機能要求を考慮したフィーチャモデル

3.4 プロダクトラインアーキテクチャの定義

整理した非機能要求と、非機能要求の実現に必要なアーキテクチャのコンポーネントとその配置を基に、プロダクトラインアーキテクチャを定義する。フィーチャモデルとの追跡性を確保するために、非機能要求の実現に必要なコンポーネントをアスペクトとして設計する。

耐故障性を実現するアスペクトの設計を例として挙げる。サービスを冗長化することで、耐故障性を実現することが可能である。耐故障性の実現に必要なコンポーネントとして、Fault であるかの判定とサービスへの通信の制御を担う Decider が挙げられる。サービスを実現する Web Service、および Web Application が Decider を介して通信を行なうことで、耐故障性を実現する。耐故障性の実現に必要なコンポーネントをアスペクトとして分離する。Web Service と Web Application 間の通信を仲介する形でアスペクトを織り込むことで、耐故障性を実現する。

図4に設計したプロダクトラインアーキテクチャを示す。

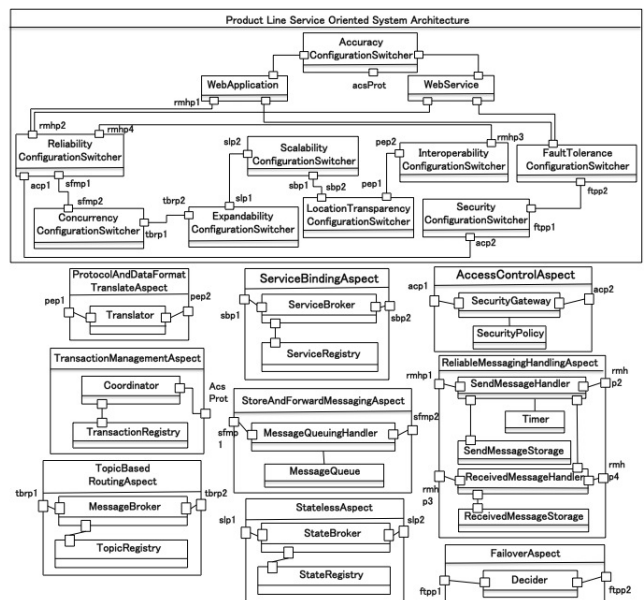


図4 プロダクトラインアーキテクチャ

図5にセキュリティと並行性を例に挙げて仕様モデルと非機能要求を実現するアスペクトの対応関係の概要を示す。非機能フィーチャと、非機能要求を実現するアスペクトとが対応付く。各アスペクトは、サービスを実現するWeb Service、およびWeb Applicationに対して織り込むことで、非機能要求を実現する。

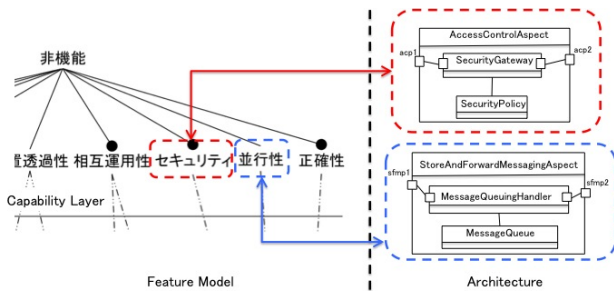


図5 対応関係の概要

4 ガイドラインを用いた歯科レセプト処理システムのアーキテクチャ設計

4.1 歯科レセプト処理システムの仕様の作成

SOAに基づく歯科レセプト処理システムに対する要求を分析し、フィーチャモデルを作成した。作成したフィーチャモデルを図6に示す。SOAに基づくシステムアーキ

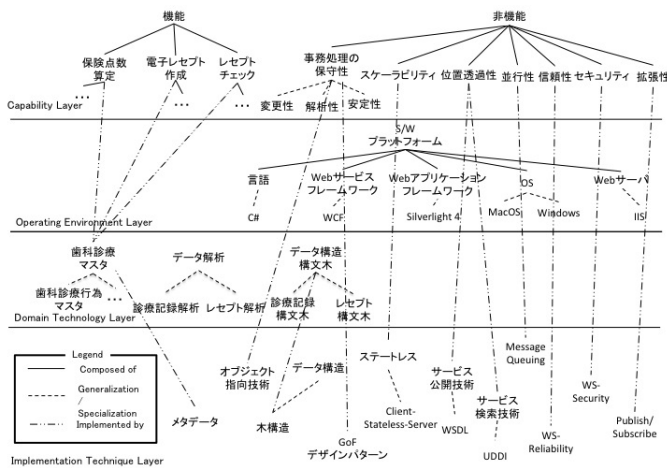


図6 歯科レセプト処理システムのフィーチャモデル

テクチャで実現すべき非機能要求として、位置透過性、スケーラビリティ、並行性、拡張性、セキュリティ、信頼性が挙げられる。

4.2 システムアーキテクチャの設計

定義したガイドラインを用い、SOAに基づく歯科レセプト処理システムアーキテクチャを設計した。ガイドラインを用い、位置透過性、スケーラビリティ、並行性、セキュリティ、信頼性の実現に必要なコンポーネントを追跡することができた。求められる非機能要求を実現するアスペクトを、歯科レセプト処理システムのサービス(レセプトチェック、保険点数算定、電子レセプト作成)

を実現するWeb Service、およびWeb Applicationに対し織り込むことで、求められる非機能要求を満たすSOAに基づく歯科レセプト処理システムアーキテクチャを設計することができた。図7に設計したSOAに基づく歯科レセプト処理システムアーキテクチャを示す。

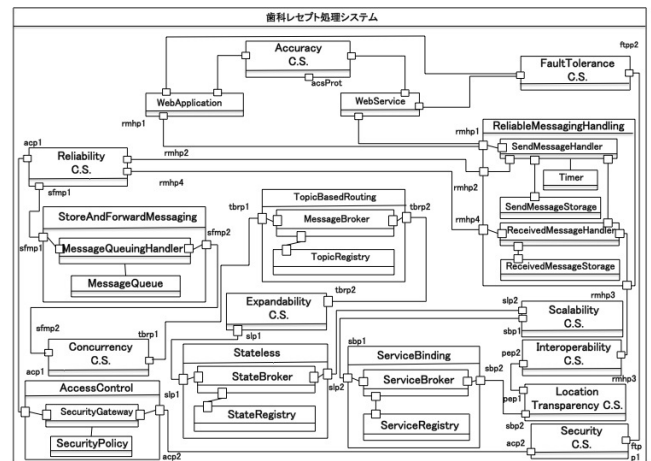


図7 歯科レセプト処理システムアーキテクチャ

5 考察

5.1 ガイドラインに関する考察

SOAに基づくシステムにおける非機能要求を整理し、フィーチャモデルと非機能要求を実現に必要なアーキテクチャのコンポーネントを定義した。定義したフィーチャモデルとプロダクトラインアーキテクチャとの追跡性を確保することで、システムアーキテクチャ設計のためのガイドラインとしてそれらを定義した。技術者は、ガイドラインを利用することで、非機能要求の実現に必要なアーキテクチャのコンポーネントを追跡することができる。よって、アーキテクチャの一定の品質の保証と、アーキテクチャ設計に要する工数の削減が見込める。このことから、定義したガイドラインが、非機能要求を満たすSOAに基づくシステムアーキテクチャ設計の支援に寄与していると考えられる。

提案したガイドラインを用いて歯科レセプト処理システムのアーキテクチャを設計した。歯科レセプト処理システムの仕様モデルを作成し、必要となるアーキテクチャのコンポーネントの配置を追跡する事で、非機能要求を満たすシステムアーキテクチャの設計ができることを確認した。よって、非機能要求を表すフィーチャとアスペクトとの対応付けが妥当であること考える。

5.2 追跡性確保の方法の妥当性

仕様モデルとアーキテクチャとが扱う関心事の違いを、アスペクト指向技術を用いてアーキテクチャ設計をおこない関心事を揃えることで、追跡性を確保した。関心事を揃える代替方法として、アーキテクチャで扱う関心事を考慮し、仕様モデルの意味や構文を拡張する方法が考えられる。この方法の問題点として、記法を拡張することで仕様モデルの本質を損なうことや、仕様モデルの作

成がアーキテクチャ設計と同意義になってしまうことが挙げられる。本研究のアプローチでは、仕様モデルの意味や構文の拡張をおこなわないことから、仕様モデルの本質を損なうことはない。仕様モデルにおける関心事をアスペクトとしてアーキテクチャを設計することから、システムの支配的分割に影響を与えることはない。仕様モデルとアーキテクチャ、それぞれの本質を損なうことなく追跡性を確保することができた。よって、本研究の追跡性確保の方法は妥当であると考えている。

5.3 追跡性確保の方法の一般性

本研究で行なった仕様モデルとアーキテクチャとの追跡性確保の方法を一般化し、手順を次に示す。

1. 仕様モデルで扱う関心事を整理
2. アーキテクチャで扱う関心事と同粒度の仕様モデルで扱う関心事を特定
3. アーキテクチャの支配的分割に対し横断する仕様モデルの関心事をアスペクトとして分離
4. 仕様モデルとアーキテクチャを対応付け

組込みシステムを対象とし、仕様モデルに FORM で提案されてるフィーチャモデルを用いた場合に、追跡性を確保することを考える。組込みシステムのアーキテクチャで扱う関心事と同粒度の関心事は、機能、非機能、およびオブジェクト（ハードウェアプラットフォーム）である。一般に組込みシステムの支配的分割はオブジェクト指向であり、機能と非機能が横断的に関心事となりうる。オブジェクト指向で設計したアーキテクチャを基に、機能、および非機能の視点からシステムを分割し、アスペクトとして分離する。これにより組込みシステムにおいても、アスペクト指向技術を適用することで、仕様モデルとアーキテクチャとの追跡性の確保が可能となると考える。組込みシステムにおけるフィーチャモデルとアーキテクチャの対応関係の概要を図 8 に示す。

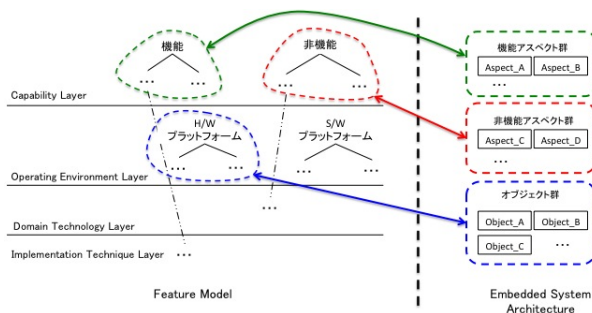


図 8 組込みシステムにおける対応関係の概要

6 関連研究

プロダクトラインの構築にアスペクト指向技術を適用する研究が行われている。Oldevik は、ドメインや何をアスペクトとするかには言及せず、アスペクト指向技術がプロダクトラインの構築に有効であることを論じている [4]。プロダクトラインアーキテクチャにおける変動体を

アスペクトとして設計することで、仕様に応じたアーキテクチャの構成要素の入れ替えが可能となる。

本研究は、プロダクトライン構築におけるアスペクト指向技術の適用に関するケーススタディとなる。我々は、エンタープライズ系システムにおける非機能要求をアスペクトとして扱い、プロダクトラインアーキテクチャを設計した。これにより、エンタープライズ系システムを対象とした場合において有効であることが確認できた。

7 おわりに

本研究では、非機能要求を満たす SOA に基づくシステムアーキテクチャ設計のためのガイドラインを定義した。PLSE の考えに基づき、非機能要求を考慮し、仕様モデルとアーキテクチャを定義した。アスペクト指向技術をアーキテクチャ設計に適用することで、仕様モデルとアーキテクチャとの追跡性の確保を容易にした。定義した仕様モデルとアーキテクチャとの追跡性を確保することで、それらをガイドラインとして定義し、システムアーキテクチャ設計の支援を実現した。

参考文献

- [1] D. Georgakopoulos, and M. Papazoglou, “Overview of Service-Oriented Computing,” *Service-Oriented Computing*, pp. 1-28, 2008.
- [2] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J. Loingtier, and J. Irwin, *Aspect-oriented programming*, Lecture Notes in Computer Science, 1997.
- [3] J. Webber, and S. Parastatidis, “Realizing Service-Oriented Architectures with Web Services,” *Service-Oriented Computing*, pp. 51-81, 2008.
- [4] J. Oldevik, “Can Aspects Model Product Lines?,” *Internal Conference on Aspect-Oriented Software Development*, vol. 2, pp. 1-8, 2008.
- [5] K. C. Kang, S. Kim, J. Lee, K. Kim, G. J. Kim, and E. Shin, “FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures,” *Annals of Software Engineering*, vol. 5, no. 1, pp. 143-168, 1998.
- [6] K. Pohl, G. Bockle, and F. Linden, *Software Product Line Engineering Foundations, Principles, and Techniques*, Springer-Verlag, 2005.
- [7] L. Bahler, F. Caruso, C. Chung, B. Falchuk, and J. Micallef “Toward Configurable QoS-Aware Web Services Infrastructure,” *Service-Oriented Computing*, pp. 151-181, 2008.
- [8] M. Shaw, D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.
- [9] 社会保険診療報酬支払基金, “レセプト電算処理システム.” <http://www.ssk.or.jp/index.html/>, 2010.