

軽量サービス指向アーキテクチャ設計方法の提案と評価

池崎 崇[†] 中道 上[‡] 青山 幹雄[‡]

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

1.はじめに

Web サービスの多くは送信情報が少量の軽量メッセージ交換であり、その構成要素と特性は多様である。しかし、軽量メッセージ交換に基づくサービス指向アーキテクチャ(軽量 SOA)のモデルは未確立であるため、軽量 SOA の実現が困難である。

本稿では、軽量メッセージ交換の構成要素と属性を定義した軽量 SOA のモデルとパターンに基づく軽量 SOA の体系的な設計方法を提案する。

2.軽量 SOA のモデルとパターン

軽量メッセージ交換は構成要素とその組み合わせによるアーキテクチャの性質(属性)が多様である。

本稿では、図 1 に示すように構造と挙動の視点から構成要素と属性を定義する。構造の視点では、軽量メッセージ(M)をメッセージ構造(S)と軽量メッセージの属性(A_M)の対として定義する。挙動の視点では、メッセージ交換(ME)をメッセージ挙動(B)とメッセージ交換の属性(A_{ME})の対として定義する。

また、構造と挙動の属性を集約した軽量メッセージ交換の属性(A)を定義し、軽量メッセージ交換の属性モデルを提案する。階層モデルを用いて、各階層に関連する属性を定義する。さらに、軽量 SOA を体系的に設計するために、属性に基づいて構成要素を組み合わせたパターン(ABAP: Attribute-Based Architectural Pattern)を提案する。

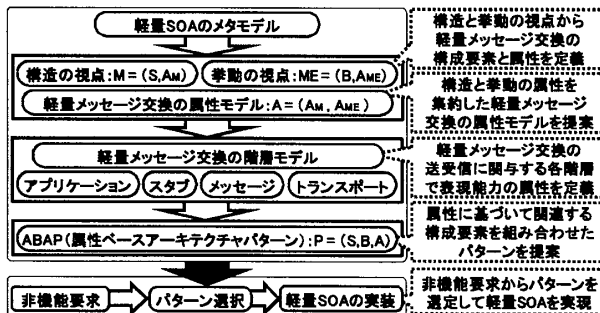


図 1: 軽量 SOA のモデルとパターン

2.1. 軽量 SOA のメタモデル

軽量 SOA のメタモデルを図 2 に示す。SOA を拡張して、メッセージ交換に軽量メッセージ交換を用いた軽量 SOA を定義する。また、軽量メッセージ交換の構成要素と属性を集約を ABAP として定義し、属性に基づく軽量 SOA をパターン化する。

Design Method of Lightweight Service-Oriented Architecture.
[†] Takashi Ikezaki, Graduate School of Mathematical Sciences and Information Engineering, Nanzan University.
[‡] Noboru Nakamichi, Mikio Aoyama, Faculty of Mathematical Sciences and Information Engineering, Nanzan University.

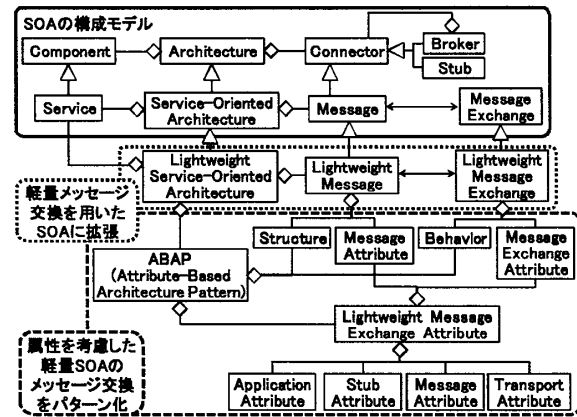


図 2: 軽量 SOA のメタモデル

2.2. 軽量メッセージ交換の属性モデル

軽量メッセージ交換の構成要素と属性を構造と挙動の視点から定義する。軽量 SOA を属性に基づいて設計するために、図 3 に示すように構造と挙動の属性を集約した属性モデルを提案する。図 3 の属性間には関連があると考えられる。

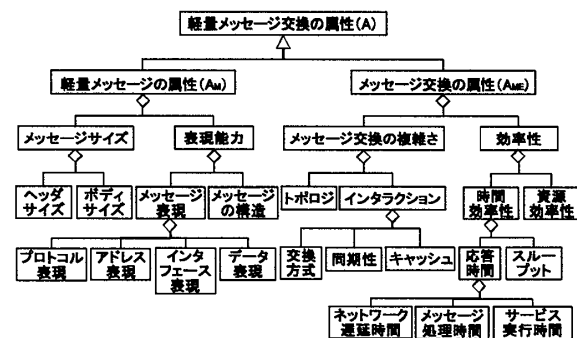


図 3: 軽量メッセージ交換の属性モデル

2.3. 軽量メッセージ交換の階層モデルとパターン化

前節で定義したモデルの属性間の関連を図 4 に示す。図 4 から、表現能力に関する属性は多くの属性に関与するため、表現能力に関する属性を詳細に定義する必要がある。階層モデルを用いて軽量メッセージ交換の送受信に関与する各階層で表現能力の属性を定義する。

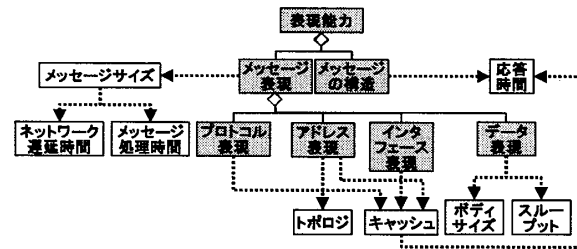


図 4: 属性の関連図

(1) 階層モデルによる属性の定義

表 1 に示すように、SOA のメッセージ交換は 4 階層に分割可能である [1]。階層モデルを用いて各階層の性質に基づき関連する表現能力の属性を詳細に定義する。

表 1: 階層モデルと軽量メッセージパターン

階層 (レベル)	属性	構成要素	軽量メッセージパターン			
			SOAP	POX	REST	Ajax
アプリ ケーション	データ 表現	データ型	*	*	単純 データ型	*
		ペイロード フォーマット	SOAP +XML	XML	*	*
スタブ	インタ フェース 表現	インタフェース 記述	WSDL	WSDL 2.0 WADL	WSDL 2.0	WSDL
		メソッド	*	*	HTTP メソッド	HTTP メソッド
メッセージ	メッセージ 表現	メッセージ の構成	ヘッダ+ エンベロープ	ヘッダ+ ボディ	ヘッダ+ ボディ	ヘッダ+ ボディ
		エンド ポイント	処理 プロセッサ	サービス	サービス	処理 プロセッサ
トランス ポート	プロトコル表現 インタフェース 表現	プロトコル	*	HTTP	HTTP	HTTP
		HTTP メソッド	POST/ GET	GET/ POST	*	*

(2) 軽量メッセージパターン

表 1 に各階層の属性に関連する構成要素とそのパラメータを示す。属性に基づいて表 1 に示す構成要素のパラメータを選定し、4 つの主要な軽量メッセージパターンを抽出する。各パターンには、固定と可変のパラメータが存在する。可変パラメータは、非機能要求や条件に基づいて決定する。

(3) 軽量メッセージパターンの制約条件

表 1 の軽量メッセージパターンは、他の属性に関連する構成要素の制約条件があり、特にメッセージ交換に関する制約条件が存在する。表 2 は、トポロジとインタラクションの属性に関連する構成要素の制約条件である。この制約条件に基づいてメッセージ交換モデルを決定する。

表 2: 軽量メッセージパターンの制約条件

トポロジ	インタラクション		軽量メッセージパターンの制約条件			
	同期性	交換形態	SOAP	POX	REST	Ajax
直接通信	同期	プル型 プッシュ型	○	○	○	○
	非同期		非同期 APIを利用	非同期 通信モデルが 必要	○	○

2.4. 属性ベースアーキテクチャパターン(ABAP)

属性に基づく軽量 SOA を体系的に設計するために、属性と属性に関連する構成要素の組み合わせのパターン(ABAP)を定義する。属性モデルの各属性(A)と関連する表現能力の属性に基づいて階層モデルで選定した軽量メッセージパターン(S)、軽量メッセージの制約条件から決定したメッセージ交換モデル(B)を組み合わせる。

3. 軽量 SOA の設計方法

属性に基づく軽量 SOA 設計は、非機能要求と条件を入力とし、図 4 に示す属性と非機能要求の関連から軽量 SOA の属性を決定する。この属性に基づき、表 1 と表 2 から定義される ABAP を選択し、軽量 SOA のパターンを決定する。軽量 SOA のパ

ターンに基づいて軽量 SOA を実装する。

この軽量 SOA の設計方法を、Ajax を例題として図 5 に示す。非機能要求から属性を定義し、属性と条件に基づいて構成要素を段階的に絞り込むことで、属性に基づく軽量 SOA が実現可能となる。

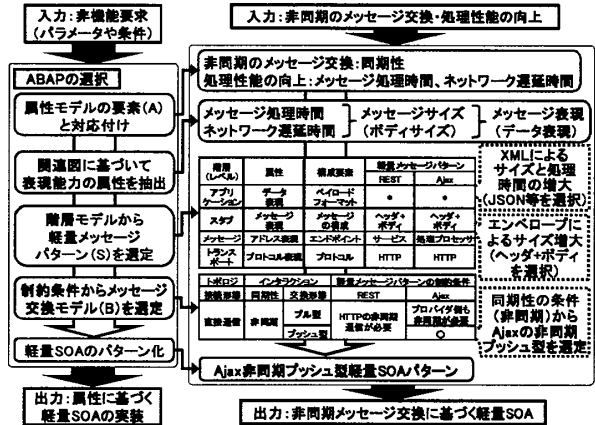


図 5: Ajax パターンに基づく軽量 SOA の実現

4. 軽量 SOA 設計方法の評価

提案した軽量 SOA の設計方法により、Web サービス開発者は様々な属性を考慮した Web サービスアーキテクチャを体系的に設計可能となる。

また、図 5 で示した Ajax 非同期プッシュ型軽量 SOA パターンのプロトタイプを Apache Axis2 を用いて試作し、他のパターンと比較してメッセージのサイズと送信時間の減少を確認した。このパターンは非同期で、かつリソース制約が厳しいシステムの実現に有効である。

5. 関連研究

REST(REpresentational State Transfer)の技術要素を適用した Web サービスと SOAP/WSDL による Web サービスの構成要素を概念と技術の観点から比較し、アーキテクチャ決定要素を抽出した研究がある [2]。しかし、抽出した要素に基づくアーキテクチャの体系的な設計方法は示されていない。

6. まとめ

本研究では、構造と挙動の視点から軽量メッセージ交換の構成要素と属性を定義し、軽量 SOA の属性モデルを提案した。また、多くの属性に関連する属性を SOA の各階層で詳細に定義した。さらに、その属性と構成要素を組み合わせ属性ベースのパターン(ABAP)を定義し、例題を用いて ABAP に基づく軽量 SOA の設計方法を示した。

参考文献

- [1] 森 晃 ほか, 非同期メッセージ交換のモデルとパターンに基づく非同期サービス指向アーキテクチャ設計方法. 情報処理学会論文誌, Vol. 48, No. 8, Aug. 2007, pp. 2566-2577.
- [2] C. Pautasso, et al, RESTful Web Services vs. "Big" Web Services, Proc. WWW 2008, Apr. 2008, pp. 805-814.