

SOA に基づく SaaS 統合アーキテクチャの提案

近藤 洋介[†] 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 情報理工学部 ソフトウェア工学科[‡]

1 はじめに

オンプレミスアプリケーションの一部機能を SaaS に移行するシステム構成が増加している。しかし、オンプレミスアプリケーションと SaaS 間でデータ統合の方法が確立されていない。本稿では、データ統合を実現するため、SOA(Service-Oriented Architecture)に基づく SaaS 統合アーキテクチャを提案する。

2 SaaS 統合の問題

現状の SaaS 統合モデルを図1に示す。このモデルでは、オンプレミスアプリケーションと統合する SaaS の数に比例して専用のコネクタを必要とする。また、オンプレミスアプリケーション側は専用のコネクタに対応したインタフェースに変更しなければならない。このように統合が複雑となりコストが増加する。

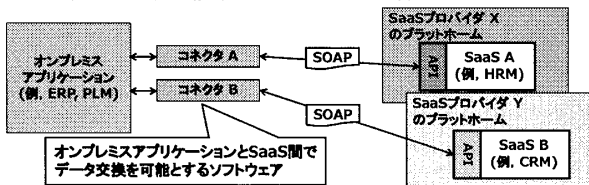


図1: SaaS 統合モデル

3 アプローチ

統一的なインタフェースから複数の SaaS とのデータ統合を可能とするアーキテクチャの実現を目指す。そこで、サービスの柔軟な組み合わせを実現している SOA のビジネスプロセスに着目し、従来の SaaS 統合モデルを拡張する。さらに、コネクタをサービスとして捉え、コネクタや他のサービスをビジネスプロセス上で組み合わせることで、様々な SaaS との統合を容易に実現するアーキテクチャを提案する。

4 SOA に基づく SaaS 統合アーキテクチャ

4.1 SaaS の特性

SOA で想定している Web サービスと SaaS を比較し、データ統合における SaaS の特性を以下に示す。

- (1) インタフェースとメッセージングの観点: Web サービスと同様に標準技術(WSDL, SOAP, JSON)を用いる。また、SaaS ごとに提供する API が異なる。
- (2) データモデルの観点: Web サービスとは異なり、構成メタデータをカスタマイズすることでテナント毎に異なるデータモデルを提供する。そのため、API を利

用して間接的に SaaS のデータベースへアクセスするには、テナント毎の識別情報とデータモデルを知る必要がある。

- (3) 機能粒度の観点: Web サービスと同様に機能粒度は API 単位となり、SaaS が提供する API に依存する。そのため、オンプレミスアプリケーションで提供していた機能とは粒度が異なる。

表 1: Web サービスと SaaS の比較

	Web サービス	SaaS
利用形態	API を用いて間接的に利用	API を用いて間接的に利用 Web ブラウザ上で直接利用
機能	機能の変更はない	間接利用: API は変更しない 直接利用: 構成メタデータを用いることで、テナント毎に異なる機能を提供
データモデル	データモデルの変更はない	構成メタデータを用いることで、テナント毎に異なるデータモデルを提供
インタフェース記述	WSDL	WSDL
メッセージング仕様	SOAP, JSON	SOAP, JSON
ビジネスプロセスとのインタラクション	同期型(リクエストレスポンス) 非同期型(一方向, コールバック)	非同期型(リクエストレスポンス) 非同期型(一方向, コールバック)
ビジネスプロセスからみた機能粒度	API 単位	API 単位

4.2 SaaS 統合アーキテクチャ

4.1 節で示した SaaS の特性と SOA に基づいて拡張した SaaS 統合アーキテクチャを図 2 に示す。

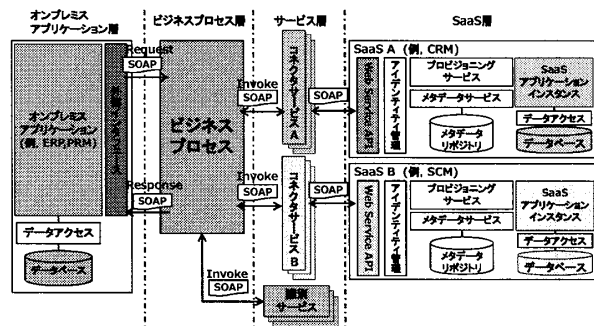


図2: SaaS 統合アーキテクチャ

提案するアーキテクチャは、オンプレミスアプリケーション層、ビジネスプロセス層、サービス層、SaaS 層の 4 層で構成する。ビジネスプロセス層により統一的なインタフェースによる統合を可能とし、サービス層で各 SaaS のインタフェース(API)の違いを吸収する。また、ビジネスプロセスでサービスを組み合わせることで機能粒度の違いを吸収する。

サービス層で提供するコネクタサービスの粒度は、SaaS の API 単位とする。これにより、API の機能に適したインタラクション(同期, 非同期)が選択できる。さらに、テナント単位で識別サービスを用い、テナント毎の識別情報や API コールに必要な情報を提供する。コネクタサービスを呼び出す前に、識別サービスから

SaaS Integration Architecture Based on SOA

[†] Yosuke Kondo, Graduate School of Mathematical Sciences and Information Engineering, Nanzan University.

[‡] Noboru Nakamichi, Mikio Aoyama, Department of Software Engineering, Nanzan University.

必要な情報を取得することで、構成メタデータのカスタマイズによるデータモデルの変更に対応できる。

4.3 SaaS 統合アーキテクチャの振る舞い

提案アーキテクチャで拡張したビジネスプロセス層、サービス層の振る舞いを図 3 に示す。

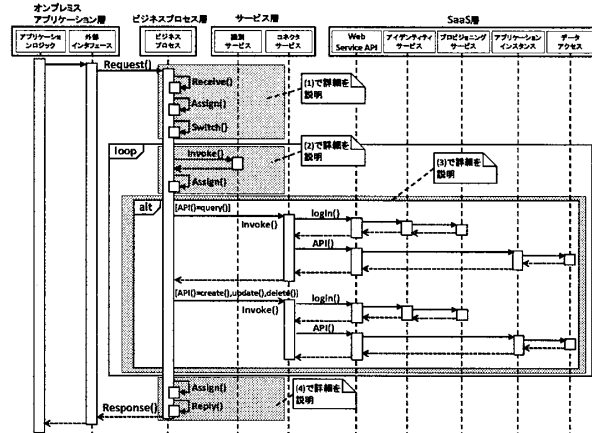


図 3: SaaS 統合アーキテクチャの振る舞い

(1) ビジネスプロセスの初期処理: リクエストメッセージを受信する。受信した値を元に、統合する SaaS に対応したビジネスプロセスを起動する。

(2) インタフェースの差異調整: テナントの識別サービス呼び出し、テナント毎の識別情報と SaaS の API コールに必要な情報を取得する。

(3) SaaS とのデータ統合: コネクタサービスを SaaS の API の振る舞いに応じて、同期、または、非同期で呼び出す。コネクタサービスは、識別サービスから取得した情報を元にテナントの認証を行い、SaaS とのセッション情報を得る。セッション情報を元に API コールを用いて SaaS のデータベースへアクセスする。

(4) SaaS 統合による処理結果の返信: ビジネスプロセスで統合した一連の SaaS の処理結果を返信する。

また、(2)、(3)の処理を必要に応じて繰り返すことで、機能粒度の違いに対応する。

4.4 メタデータの定義

本稿では、ビジネスプロセスに BPEL を用いることとする。提案アーキテクチャを様々なドメインに適用するため、BPEL の抽象プロセスと実行可能プロセスをメタデータとして定義する。メタデータは XML スキーマを用いて定義し、図 4 にそのクラス図を示す。

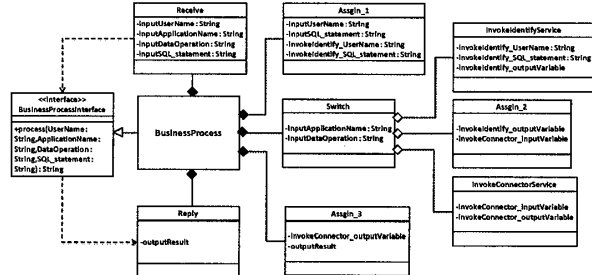


図 4: ビジネスプロセスのメタデータ

5 プロトタイプによる検証

提案アーキテクチャに基づいてプロトタイプを開発し、従来の SaaS 統合モデルと比較した。

5.1 実装

表 2 に開発したプロトタイプの仕様を示す。

表 2: プロトタイプの仕様

	オンプレミスアプリケーション	ビジネスプロセス	識別サービス	コネクタサービス	SaaS[1]
概要	Oracle が提供する WebLogic テストクライアントを利用	BPEL を用いて開発 (同期型)	サービスの粒度をテナント単位で開発 (同期型)	サービスの粒度を API 単位で開発 (同期型, 非同期型)	Salesforce.com を利用して開発
配置	Oracle BPEL Process Manager	Tomcat5.5 Axis2	Tomcat5.5 Axis2	Tomcat5.5 Axis2	Salesforce 社のサーバ
機能	・ビジネスプロセスの呼び出し ・応答時間の計測	・Web サービスの呼び出し ・プロセス制御	・テナント別の識別情報を提供 ・各 API コールに必要な情報を提供	・query() (同期型) ・create() (非同期型) ・update() (非同期型) ・delete() (非同期型)	各 API コールに応じた処理
インタフェース記述	WSDL	WSDL	WSDL	WSDL	WSDL
メッセージング仕様	SOAP	SOAP	SOAP	SOAP	SOAP

5.2 実行結果

従来の SaaS 統合モデルと提案アーキテクチャの応答時間をそれぞれ 10 回計測した。平均応答時間を表 3 に示す。ビジネスプロセスの処理によるオーバーヘッドは平均 650(ms)であった。

表 3: 応答時間の計測結果

	利用したコネクタサービスの機能	コネクタサービスとのインタラクション	Web Service API	平均応答時間 (ms)
ビジネスプロセス層なし	データ取得機能	同期型 サービス呼び出し	query()	1564(ms)
ビジネスプロセス層あり	データ取得機能	同期型 サービス呼び出し	query()	2211(ms)

6 評価

本稿で提案したアーキテクチャは、統一的なインタフェースから複数の SaaS とのデータ統合を実現する。また、ビジネスプロセスを用いることで、新しい SaaS との統合が容易なシステム構成となっている。プロトタイプを用いたビジネスプロセスの処理オーバーヘッドの評価から、提案アーキテクチャは妥当といえる。

7 関連研究

複数のコネクタ機能を提供する Integration as a Service Provider を利用して SaaS 統合を実現する研究がある[2]。しかし、Integration as a Service Provider がサポートしている特定の SaaS としか統合できない。

8 まとめ

SOA に基づく SaaS 統合アーキテクチャを提案した。SaaS の特性を示し、構造と振る舞いの点からアーキテクチャを定義した。また、プロトタイプを用いて従来の SaaS 統合モデルとの比較を示した。

今後の課題として、提案アーキテクチャを用いた複数の SaaS とのデータ統合を評価する。

参考文献

[1] C. Roth, et al., Force.com Cookbook, Salesforce.com, 2008, <https://www.salesforce.com/jp/form/offer/cookbook.jsp>.
 [2] H. Hai, et al., SaaS and Integration Best Practices, FUJITSU Sci. Tech. J., Vol. 45, No. 3, Jul. 2009, pp. 257-264.