

課金機能 API の SOAP 化の実装と考察<sup>1</sup>上野 正巳, 飯田 恭弘, 阿川 雄資<sup>2</sup>NTT 情報流通プラットフォーム研究所<sup>3</sup>

はじめに

インターネットの標準を組み合わせることでプログラマブルなアプリケーションコンポーネントへのアクセスを実現する Web サービスが近年話題となっている。我々は Web サービスの構築及び運用における課題の検証を目的とした実証実験 [1] に参加し、課金プラットフォーム InfoBilling Lite [2] (以下課金 PF と略記) の API の SOAP 化を行った。本稿ではその実装で得られた知見について考察する。

## システム構成

システムの構成を図 1 に示す。各 Web サービスの連携はサービス連携機構 eCo-Flow [3][4] が行う。サービス全体のフローと、サービス利用者のための画面の遷移がここで記述される。また、プラットフォーム連携規定 [5] (以下 PF 連携規定と略記) を補助する PF 連携機構も eCo-Flow 上にインプリメントされ、課金 PF はその制御対象となる。サービスアプリケーションが課金 PF を Web サービスとして呼び出す際には SOAP クライアント部分が UDDI を参照する。

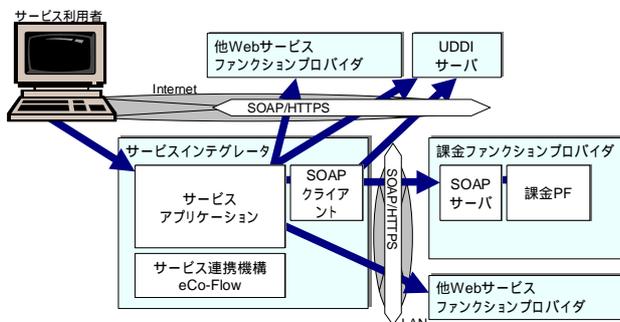


図 1 システム構成

## 課金プラットフォーム

課金 PF は課金コア機能として基本的なユーザ管理機能、物販などで利用する随時課金機能、会費等で利用する周期課金機能、課金データを蓄積するタンキング機能、外部の決済代行システムとの接続を行うインターフェース機能を持つ。課金 PF の各機能呼び出す API はアダプタ機能

が提供し、PF 連携規定に準拠した Java の RMI-I/F を提供している。SOAP 化は RMI の I/F を利用する上位アプリケーションの互換性を確保するために RMI-API を変更せず、図 2 のようにアダプタ機能とコア機能の間の RMI-I/F の部分に SOAP/HTTP の層を挟み込む形で対応を行った。

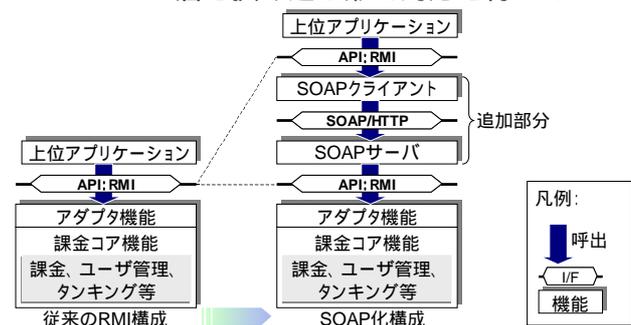


図 2 課金 PF の SOAP 対応

## SOAP 化の実装と考察

Web サービスが想定している理想的な利用形態の特徴は以下である。

- ・ マルチベンダのサービスの統合
- ・ 分散環境
- ・ インターネット上をサービスオブジェクトが流通する
- ・ 片道ではなく相互作用を与える Web サービスの提供

これらの特徴を考慮して設計を進める際に得られた、システム設計上考慮しなければならない事項を以下に述べる。

## SOAP-I/F の粒度

もともと課金 PF はローカルシステムでダウンロード販売等のサービスアプリケーションが動き、そこからの課金要求を処理する事を想定して設計されている。このため、課金 PF の API は比較的粒度の小さい低レベルの API を提供している。Web サービスへの対応に際してはこの API の粒度を変えず、そのまま SOAP 化を行ったが、広域分散も可能な Web サービスのファンクションプロバイダとしては、業務フローに対応して複数の処理を 1 回の呼び出しで行えるような実装が望

<sup>1</sup> Implementation and Study of SOAP-nized Billing system API

<sup>2</sup> Masami UENO, Yasuhiro IIDA, Yuji AGAWA

<sup>3</sup> NTT Information Sharing Platform Laboratories

ましい。例えば会員登録時に会費を徴収するような業務の場合、課金 PF では会員登録と課金の API は別々になっているが、これらを図 3 の右図のように 1 回の API で呼び出し可能になれば、SOAP 化に伴うオーバーヘッドを減らす事が出来、性能面での向上が期待できる。他の課金 API や業務においても同様の検討が必要になる。

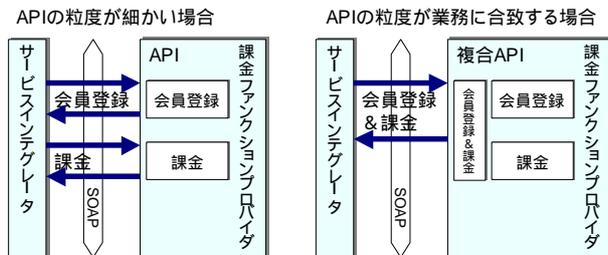


図 3 API の粒度の違いによるオーバーヘッド

### SOAP ツール間互換性

図 2 左図で示した構成の課金 PF が RMI で提供している従来の API のパラメータでは Map 型、Vector 型の SOAP では使用できないデータ型を利用している。このため、これらを SOAP で利用可能な型に変換して呼び出しを行う必要がある。一般に SOAP サーバやクライアントを作成する際には、Apache-AXIS[6]等の SOAP ツールを用いてインプリメントを行うが、使用するツール間で受け渡し可能な型に非互換の部分があるため、前記のような型変換を行う際にはなるべく多くの SOAP ツールで問題なく利用できる事が望ましい。このため、JAX-RPC[7]などのメジャーな SOAP ツールとの間で、型と例外伝播の互換性を確認し、問題のない方法で SOAP-I/F で利用する型を決定した。

### WSDL の公開・非公開と記述レベル

サービスインテグレーター一般に公開する読み出しのみの機能の Web サービスであれば WSDL を公開しても問題のない場合が多いが、課金機能のように本来契約が必要なサービスを提供するような場合、セキュリティ面から WSDL を一般に公開するのが望ましいかどうか詳細に検討する必要がある。本実験では WSDL は一般公開せずに実施した。

また、WSDL の記述レベルに関して検討の必要がある。課金 PF の SOAP 化に際しては Java の Map 型を SOAP の Struct 型に変換して利用しているために、その内部の構造まで記述するかどうか決める必要があるが、これはセキュリティ面や、クライアントがダイナミックにサービスをバインドするかどうかで要件が異なってくる。

### UDDI 検索のタイミング

UDDI 検索のタイミングは、性能面や運用上の制約から決定する必要がある。SOAP-I/F の呼び出し毎に毎回検索する、一度検索したものをある程度の期間キャッシュして再利用する、または最初に 1 回検索した情報を使いまわす等の案が考えられる。

### SOAP-I/F のセキュリティ

SOAP-I/F のセキュリティの要件に応じて、I/F 自体を制限無く一般公開すべきか、それとも認証を行った上で暗号通信の必要があるのか、インターネットを利用してデータのやりとりを行っても安全かどうか検討を行う必要がある。課金機能のように個人情報が流通する場合は認証と暗号化はそれぞれ必須、サービス提供システムはインターネット経由でのアクセスは許すが、課金や会員情報の管理系は外部からのアクセスを許さない方式が適する。

### おわりに

今回課金機能 API の SOAP 化を通して、既存 I/F を SOAP 化する上で様々な問題点や検討を要する点が明確になった。特にサービスが扱う情報の種別によりセキュリティへの要求が高くなる点は十分に注意する必要がある。また、セキュリティを高めると性能面で影響を受けるものの、サービス全体から見ると小規模な影響しか与えないことが分かった。

### 参考文献

- [1] [http://xml.fujitsu.com/jp/about/2002\\_0916.html](http://xml.fujitsu.com/jp/about/2002_0916.html) (2003/01/10 アクセス)
- [2] 阿川他, “SI/PF ビジネススタートを支援する課金決済プラットフォーム InfoBilling Lite”, NTT 技術ジャーナル 2003 年 1 月号, 電気通信協会, 2003. 掲載予定.
- [3] Hatashima, et al. “WebServices Processing Platform eCo-Flow”, SAINT-2002, IPSF and IEEE-CS, 2002
- [4] <http://pr.fujitsu.com/jp/news/2002/03/1.htm> (2003/01/10 アクセス)
- [5] 日野隆一他, プラットフォーム連携による AP 開発手法の提案, 電子情報通信学会技術研究報告, Vol.101, No.742, 電子情報通信学会, 2002
- [6] <http://xml.apache.org/axis/> (2003/01/10 アクセス)
- [7] [http://jdc.sun.co.jp/techarticles/Rev3/jax\\_rpc\\_03.html](http://jdc.sun.co.jp/techarticles/Rev3/jax_rpc_03.html) (2003/01/10 アクセス)