

# INTERNET

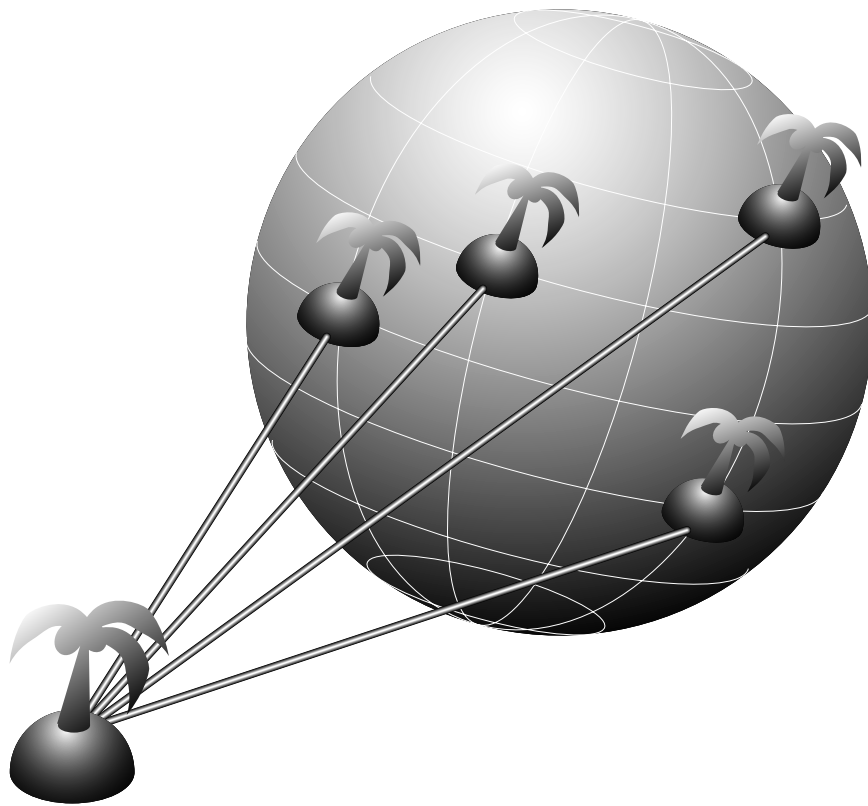
● インターネット最新テクノロジー：第46回

ウェブにおける技術革新の基礎を支える

## SOAP (Simple Object Access Protocol)

これまでのウェブサービスは、各サイトが独自のバックエンドを持ち、ばらばらにサービスを提供しているという「サービスの孤島」であった。ところが現在、インターネット上にある無数のウェブサービスをダイナミックに結合させようという技術革新が起きつつある。今回はこの革新の基礎を受け持つプロトコル「SOAP」について解説する。

岡部 恵造 株式会社 イー・ブリッジ



### 孤島化からの脱却SOAPがもたらすウェブサービスの連携

もしあなたが出張を命じられ、航空券の購入とその経費の申請をウェブサイト上で行うということになった場合、どのようにウェブを利用するだろうか。

これまでのウェブの利用法では、航空券購入のウェブサイトと、経費の申請を行うウェブサイトとの両方にアクセスして、それぞれにデータを入力しなくてはならなかった。しかし、もしこの2つのサービスが1つのサイトで提供されていれば、入力するデータ量も少なく済み、ユーザーの負担はかなり軽減されるはずだ。

このように、ウェブサイトが提供しているサービスがそれぞれサイトごとに完結している状態、いわゆる「サービスの孤島」となっていることによる不便は大きい。

このような不便を解消するのが、今回紹介するXMLベースのRPC (Remote Procedure Call) プロトコル「SOAP」だ。

先ほど挙げた例だと、経理システムとチケット会社のウェブサービスが連携することで、チケット会社が持つデータをあたかも経理システムのバックエンドかのように利用し、チケットの入手と経理への申請とを同時に行うといったことが可能になる。つまり、1度のユーザーアクションで複数のウェブサービスを利用できるようになるわけだ (図1)。

実はこれまでも、マイクロソフトのDCOMやOMGのCORBA/IIOPといったオブジェクトモデルを使うことでウェブサービスを連携させることができたが、この方法だと双方がそれぞれ同じオブジェクトモデルを利用している必要があり、限られた連携にすぎなかった。それに対してSOAPは、インターネットに接続されたすべてのクライアントやサーバーが必ず実装しているHTTPやSMTPなどの標準トランスポートプロトコルを利用するため、インターネット全体を視野にした広域での連携をもたらしてくれる。

## ジャストインタイムによる アプリケーション構築

それでは、RPCそのものの概念とそこでのSOAPの動きを解説する。RPCはネットワーク上に分散している遠隔地アプリケーションをサブルーチンのように呼び出す技術方式のことだ。API (Application Programming Interface) を用いた密結合のアプリケーションの場合は、実行する前からそれぞれのモジュールが結合しているのだが、RPCメカニズムのアプリケーションでは、インターネットに分散しているモジュールをSOAPで呼び出し、一時的に結合させて実行するため、拡張性が高い(図2)。呼び出せるウェブサービスには、マイクロソフトの「HailStorm」のようなビジネス共通の基盤となるものから、「金融決済」「在庫引当」といった個々のビジネスサービスまであり、アプリケーションがこれら多様なウェブサービスを共有する。モジュール単位でのアウトソーシングと言えるだろう。

## ウェブサービスに投げられる SOAPメッセージ

SOAPは、潜在的にはほかのさまざまなプロトコルとの組み合わせで利用できるが、SOAP1.1仕様では、HTTPとHTTP拡張フレームワークとの組み合わせで使う方法のみが定義されている。

SOAPのメッセージは、自分がSOAPメッセージであることを示すエンベロープ要素、SOAPメッセージに必要な固有のヘッダー要素、メッセージ本体であるボディー要素の3つで構成されており、全体がXMLで記述されている。図3ではHTTPリクエスト中に埋め込まれたSOAP送信メッセージと、HTTPレスポンス中に埋め込まれた応答メッセージの例を挙げている。最新の株価を調べるGetLastTradePriceというウェブサービスに「DISの株価を教えてください」とリクエストが送信され、レスポンスとして「34ドル50セント」が

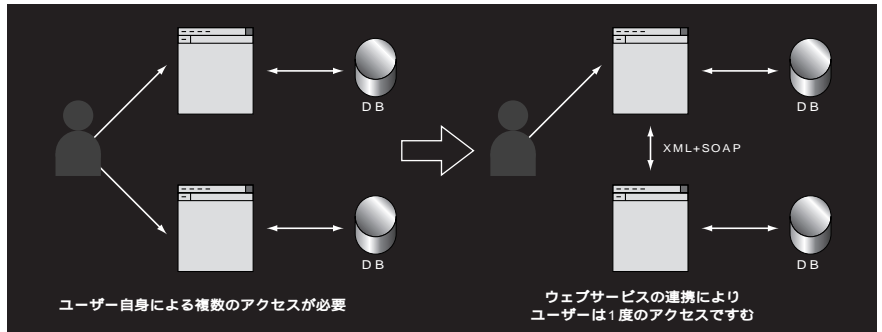


図1 ウェブサービスの連携

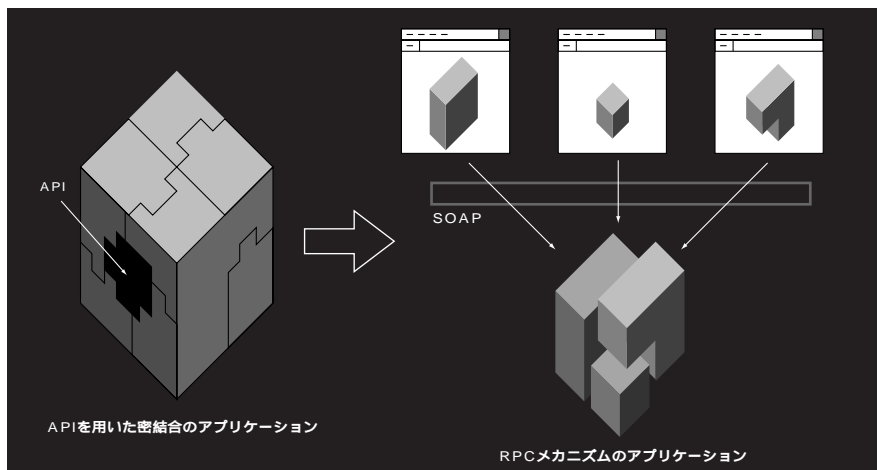


図2 RPCメカニズムのアプリケーション

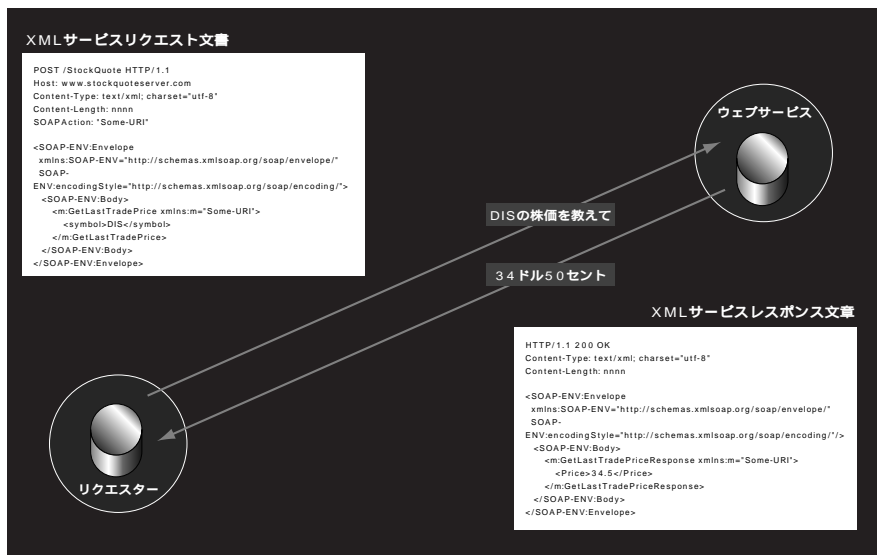


図3 SOAPメッセージを使ったやりとり



返送されているのがわかるだろう。

## SOAPでは解決できない RPCを実現する際の問題点

これまでSOAPを使ったウェブサービスの連携について解説してきたが、実際にアプリケーションの実行と同時にウェブサービスと接続するには、以下のことを実現できなければならない。

- ウェブサービスを説明する標準化された記述方法（記述言語）が存在すること。
- 必要なウェブサービスをインターネット上で画一的に見つけ出せる仕組みがあること。
- 発見したウェブサービスにアクセスする汎用的な方法が標準化されていること。
- 複数のウェブサービスを統合して、1つの意味ある機能を果たすプロセス記述言語が標準化されていること。
- 複数のウェブサービスにまたがるトランザクションの管理ができること。
- ウェブサービスへのアクセス時に高度なセキュリティが確保されていること。

これらのことは、SOAP単体では解決できない問題だ。しかし、それぞれ解決策が提示されているので以下で解説しよう。

## ウェブサービスを実現する インターネット上の三角関係

リストの から までの要件を実現させようとしているのがUDDI (Universal Description, Discovery and Integration) というインターネット用のグローバルディレクトリーの仕組みだ。

図4にあるとおり、UDDIはウェブサービスを提供する「プロバイダー」と、サービスを探して利用する「リクエスター」、その仲介をする「ブローカー」の三角関係により成立している。

まず、プロバイダーは標準化されたXMLでウェブサービスについて記述する。この記述は電話帳のディレクトリー情報によく似ており、図5にあるとおり「ホワイトページ」(ビジネス情報)、「イエローページ」(業界別のサービス情報)、「グリーンページ」(接続情報)という3種類の情報と、実際に接続する際に必要なインターフェイス情報が記載されているWSDL (Web Services Description Language) などから構成されている。このWSDLにはModelKeyと呼ばれるウェブサービスへのインターフェイス仕様を一意に識別するIDや、SOAPでアクセスする際に使用するXMLの仕様などが記述されている。

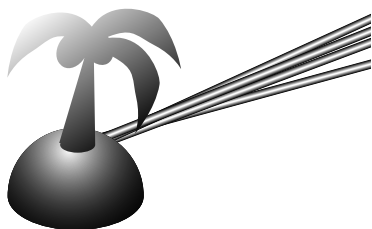
そして次に、プロバイダーはSOAPを使用してブローカーが提供するインターネット上のディレクトリーにその記述を登録する。

それに対し、ウェブサービスを求めるリクエスターは、同じくSOAPを使用してディレクトリーにある各ページを検索して、最適のウェブサービスを見つけ出す。そして、WSDLなどの接続情報をダウンロードし、UDDI対応のB2Bサーバーにインストールしてウェブサービスに接続する。

UDDIの仕組みによって、プロバイダーは多くのリクエスターに自らのサービスを利用してもらえる。そしてリクエスターは必要なウェブサービスを見つけ出してすぐに接続し利用できる。その結果、UDDIを利用することで、テレビのチャンネルを切り替えて番組を受信するかのよう、ウェブサービスもダイナミックに切り替えられる。

## 複数のウェブサービスの利用を 可能にする2つのポイント

アプリケーションを構築する際に、1つのウェブサービスへのアクセスだけで機能をすべて実現できるケースは少なく、実際には複数へのウェブサービスに一定の順番でアクセスすることが多い。そこで先ほどのリストの「プロセス定義」が必要になる。現在はブ



ロセスを記述するXMLとして、マイクロソフトのBizTalk サーバーで利用されている XLANG と IBM の WSFL ( Web Services Flow Language ) などがあり、標準化が進められているところだ。

またそのアプリケーションの途中で障害などが発生し処理を中止する場合、それまでに実行した処理もロールバックして元に戻す必要がある。冒頭に挙げた例で言えば、チケットの予約はとれたけれども、経理システムがうまく働かなかったといった場合だ。このようなときは、チケットの予約自体もなかったことにしなくてはならない。このようなトランザクションの調整を行うのが XAML ( Transaction Authority Markup Language ) であり、現在のところ開発中だ。

## ウェブサービスにおけるセキュリティの確保

のセキュリティに関しては、いくつかの重要な取り組みがある。非技術的な取り組みとしては、ウェブサービスに対する格付けや与信管理などが進められており、また技術的な分野では SOAP のデジタル署名に関する拡張の提案、XKMS ( XML Key Management Specification ) というセキュリティサービスの提案、信用状を発行する SAML ( Security Assertion Markup Language ) と言語の標準化が進んでいる。

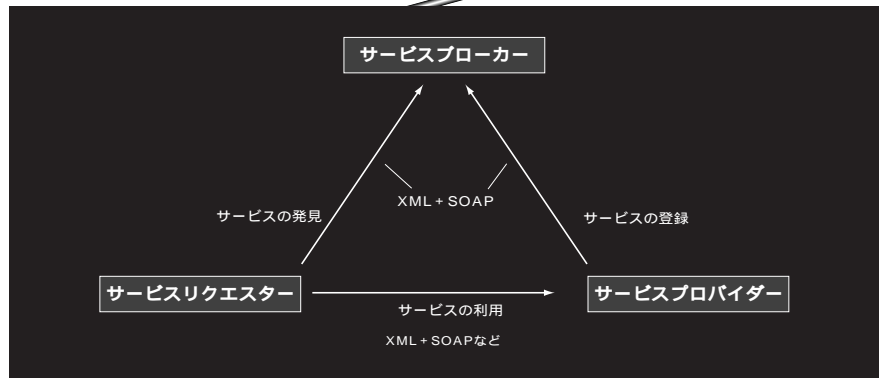
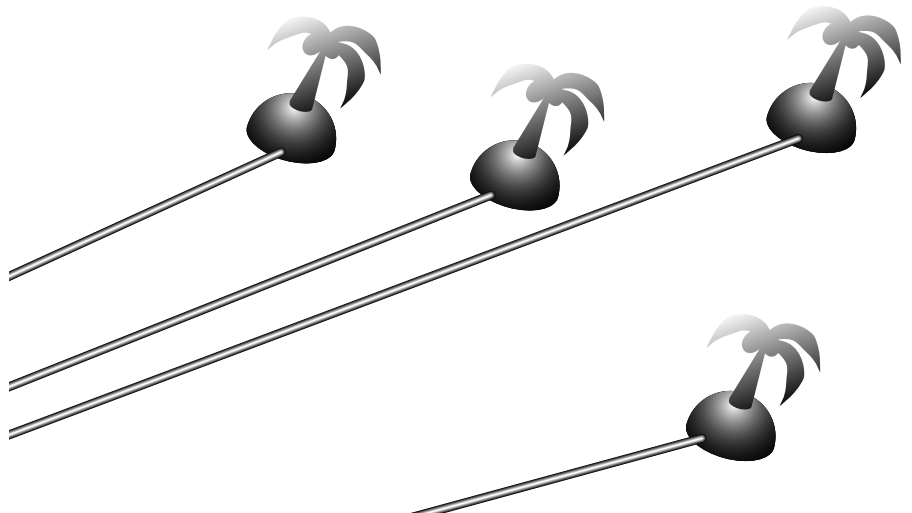


図4 UDDIの三角構造

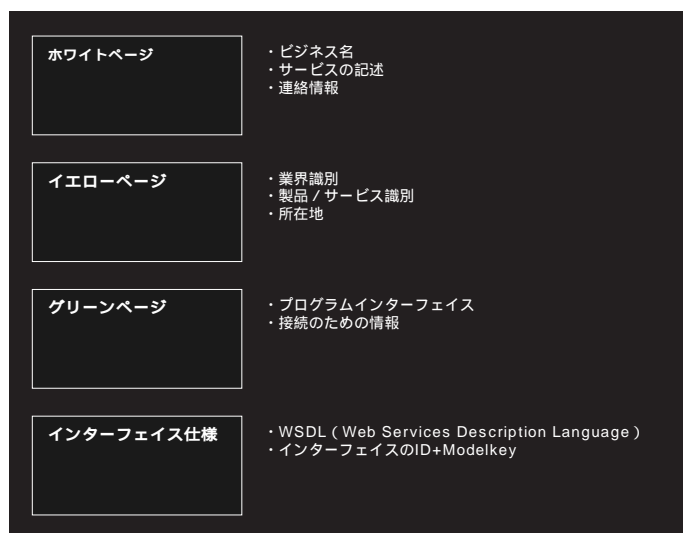


図5 UDDIのディレクトリーサービスの構造



## [インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

**株式会社インプレスR&D**

All-in-One INTERNET magazine 編集部

[im-info@impress.co.jp](mailto:im-info@impress.co.jp)