

Feature **1**

最新ソフトウェア技術の大本命を 基礎から実践まで一挙解説

本誌でも繰り返し述べてきたように、インターネットは大きな曲がり角に来ています。HTMLの不便さを乗り越えるためにXMLが生まれ、そのXMLを基盤にして「Webサービス」が登場しつつあります。今回の特集では、今後のソフトウェアアーキテクチャーの大本命、Webサービスを基礎から徹底的に解説します。

今こそ学ぶ!

Webサービス

PART 1

Webサービスで
ビジネスが変わる
栗原 潔

PART 2

IBM WebSphere Studio V4.0
によるWebサービス構築法
米持 幸寿

PART 3

SOAP Toolkit 2.0を使った
SOAPサービスの構築
田中 正造

PART 4

ソフトウェアベンダーに聞く
Webサービスへの対応状況
XML Magazine編集部

ス



PART 1

Webサービスで
ビジネスが変わる

次世代インターネットコンピューティングのすべて

Web
Services

ガートナージャパン株式会社

栗原 潔 KURIHARA, Kiyoshi

はじめに

Webサービスは、2001年のIT業界における最重要キーワードと言って良いかもしれない。しかし、Webサービスの概念そのものはさほど複雑でないにもかかわらず、今ひとつ全貌がつかめない人も多いようだ。

本稿では、Webサービスの技術的詳細よりも、その登場の必然性、インターネット・ビジネスへの影響、将来の長期的展望などにフォーカスを当てて論じていきたい。Webサービスの影響は、プログラム間通信方式が単にひとつ増えたというだけにとどまらない。Webサービスは、インターネット上のコンピューティング方式とビジネスに長期的な影響を与える重要なパラダイム・シフトなのだ。

Webサービスとは何か？

ワールドワイドのIT動向調査と予測を行なっているガートナー社は、Webサービスを単なる流行語のレベルではなく、今後のIT業界に大きな影響を与える可能性が高い重要なテクノロジーのひとつとしてとらえ、「標準プロトコルを使用し、インターネット経由でアクセス可能な疎結合型のソフトウェア構成要素」と定義している。この定義の具体的な意味については以下に述べていくが、簡単に言ってしまうと、従来からある分散オブジェクトやコンポーネントの考え方をインターネット上で展開するものと考えれば良いだろう。つまり、インターネット上の任意の場所に存在するソフト

ウェア機能を探し出し、呼び出すことによって、その機能を使用できるようにするための仕組みである。

一方、Webサービスという言葉は広く知られるようになったが、どうもその実体がつかみにくいという声も聞かれる。その要因のひとつは「Webサービス」という名称ではないだろうか。そもそも、WebサービスはWebブラウザ経由でアクセスされるとは限らない（この点は、多くの人が誤解しているようだ）。また、Webサービスの「サービス」とは「既定のインターフェイス経由でアクセスされるソフトウェア機能」という意味のテクニカル・ターム（いわば、粒度の大きいコンポーネントと思えば良い）、ASP（アプリケーション・サービス・プロバイダ）における「サービス」のようにアプリケーション機能をレンタル方式で提供するという意味、さらにはコンサルティング、SI、ハードウェア保守などの人間による付加価値作業の総称など、多くの意味を持っている。そのため、例えば、「我社のWebサービス戦略は……」という発言を聞いても、本稿で述べているテクニカル・タームとしてのWebサービスのことが、Webを使った顧客サービスのことが、まぎらわしいこと甚だしいのである。テクニカル・タームとして最適な名称が付けられたとは言いがたいが、Webサービスという用語がある程度定着してしまったため、今後別の用語が使用される可能性は低いだろう。「Webサービス」という用語を聞いた時には、頭の中で「インターネット・コンポーネント」「グローバル・コンポーネント」とでも置き換えて考えれば、よりすっきりするのではないだろうか。

なぜ、Webサービスが必要になるのか？

Webサービスは、複数の要因に影響されて必然的に登場したテクノロジーであり(図1) 将来のインターネットにおける重要な構成要素になると予測される。過去においてインターネット上でプログラム間通信を行ってきた方法を以下に挙げる。

- ・ソケット上で直接やり取りする方法
 - ・EDIなどのファイル転送ベースの方法
 - ・特定製品(ミドルウェアやアプリケーション・パッケージ)に依存したプロトコルを使用する方法
 - ・Webサーバーが送信するHTMLのストリームを直接読み取るなどのあまりエレガントではない方法
 - ・RosettaNetなどの特定業種向けに特化した方式を使用する方法
- インターネット上でプログラム同士がリアルタイムで自由にやり取りするための、オープンで業種を問わ

図1 Web T[rXoE vv



ずベンダ中立であり、確固としたアーキテクチャに基づいた仕組みは今までありそうでなかった(表1)。B2B ECやe-マーケットプレイス向けなどで企業間の連携が容易に行なえるかのように謳っている製品も、実際はWebブラウザを操作する人間を介して連携を行なうようになっていることが普通であり、企業のアプリケーション間で直接やり取りをするには、双方の企業に同じパッケージ製品を導入し、複雑な統合作業を行なう必要があることがほとんどである。Webサービスは、このインターネットの重要なミッシング・リンク(失われた鎖)を埋める存在にあたる。

なぜ、既存の仕組みでは不十分なのか？

先に、Webサービスとは、インターネット上で分散オブジェクト・コンピューティングを実現する仕組みであると述べた。それならば、CORBA/IIOP、COM/DCOM、Java/RMIなど、既存の分散オブジェクト・コンピューティング基盤と通信プロトコルを使用すれば良く、新しい仕組みをわざわざ作り出す必要性はないと、多くの人が疑問を感じるであろう。実際、これらの分散ソフトウェア基盤が提唱された初期の段階には、企業内のみならず、インターネットを使った企業間のやり取りにも使用されるであろうと期待されていたこともあった。しかし、いまだにこの期待は実現されていない。これらの分散オブジェクト基盤をインターネット上で広範に展開する上では以下のような問題点があるからだ。

① 既存の分散オブジェクト・コンピューティング基盤はファイアーウォール・フレンドリではない

DCOMなどのプロトコルをインターネット上で広範に適用するために、現実として最大の障害となるのはファイアーウォールの存在である。例えば、DCOMでは広範囲のポート番号を動的に割り当てるため、インターネット上でのやり取りにDCOMを適用するために

表1：インターネット上のアプリケーション間通信の選択肢

	リアルタイム性	オープン性	製品独立性	業種独立性	アーキテクチャの健全性
ソケット・ベース・プライベート・プロトコル		x			x
EDI	x			x	
B2B EC製品 (従来型)通信ミドルウェア			x		
HTMLストリームの直接読み取り/生成					x
Webサービス					

は、ファイアーウォールで多くのポートを開放する必要がある。これはセキュリティ的なリスクとなるため、ほとんどのネットワーク管理者はこのような変更には消極的であろう。無理を言ってもファイアーウォールの設定を変えれば済む話ではないかと思われるかもしれないが、そう単純ではない。Webサービスの世界では、特定の企業間のやり取りだけでなく、インターネット全体における自由なやり取りが対象になっているからだ。世界中のあらゆる企業とISPのファイアーウォール設定の変更を強制することは現実的ではないだろう。

② 既存の分散オブジェクト基盤には、特定製品のランタイムの存在が前提となってしまう

DCOMやIIOPなどのプロトコルの仕様は公開されているので、原理的にはいかなるソフトウェア実装によっても、これらのプロトコルを使ったやり取りが実現可能はずである。しかし、実際には、これらの方式では、独自のバイナリ形式に基づいたメッセージ交換を行なう必要上、ランタイムとしてかなり複雑なソフトウェア基盤が必要となる。結果として、IIOPでやり取りをするためには両システムにCORBA製品の存在が必要となり、DCOMでやり取りをするためには両システムにマイクロソフト製品が必要となってしまう(マイクロソフト以外のOS上でDCOMをサポートする製品も存在するが、あまり一般的とは言えない)。つまり、これらの基盤では、ハードウェア、OS、プログラミング言語レベルの中立性はある程度実現できるものの、ソフトウェアの基盤も含めた真のプラットフォーム中立性は実現しにくい。インターネットの世界では、通信を行なう対象が、従来型のパソコンやサーバーに限定されず極めて多岐にわたる。携帯電話やPDAなどのあらゆる情報機器や情報家電にまで、同じ分散オブジェクト・コンピューティング基盤を実装することは困難だろう。

Webサービス基盤に求められる要件

上記の問題点を考えれば、Webサービスの実現手段として必要とされる通信プロトコルの要件は、自ずと明らかになる。

- ① 一般的にファイアーウォールを通過する設定がされている標準的ネットワーク・プロトコルを使用する
HTTP、FTP、SMTPなどのプロトコルは、ほと

んどのファイアーウォールを通過できるように設定されている。Webサービスも、新しいプロトコルを作り出すのではなく、これらの既存プロトコルに相乗りすべきである。

② 分散オブジェクト・コンピューティング基盤からの独立性を維持する

既存の分散オブジェクト・コンピューティングで、典型的なバイナリ形式のエンコーディングを行なうと、必然的に複雑なランタイム・サポートが必要となり、特定製品への依存性が高まる。現実的なアプローチとして、もう一段上の抽象化を行ない、よりシンプルな通信方式を採用し、分散オブジェクト基盤からの独立性(そして、その結果としての真のプラットフォーム独立性)を確保する必要がある。Webサービスでは、テキスト形式で、かつ、容易にメッセージのコンテンツを解釈できるようにするために、自己記述型のメッセージ形式を採用すべきである。このテキスト形式の自己記述型メッセージという考え方は決して新しいものではなく、今までも特定業界向けのEDIで使用されたり、OSI標準のASN.1 (Abstract Syntax Notation)として規定されたりしてきた。しかし、今は自己記述型のメッセージ形式として普及することが確実なXMLがある。Webサービスの実現手段としてXMLを使わない手はないだろう。

Webサービスの重要標準：SOAP

現時点で、上記ふたつの要件を満たし、かつ、Webサービスの標準として定着する可能性が最も高いプロトコルは、言うまでもなくSOAP (Simple Object Access Protocol) である。SOAPは、XMLでエンコードしたメッセージを、HTTPなどのインターネット標準プロトコルでやり取りするというその名の通りシンプルなプロトコルである。SOAPは、マイクロソフト社およびデベロップメンター社とユーザーランド社により共同開発されていた初期段階では、XML-RPC (Remote Procedure Call) と呼ばれていたが、この名称の方がプロトコルの実体を適切に表わしているかもしれない。SOAPの仕様上、下位のネットワーク・プロトコルは特に限定されていないが、少なくとも当面の間は、HTTPないしHTTPSによる単純な要求-応答型の通信形態が中心的に使用されることになるだ

ろう。HTTP(S)を使用することで、前述のファイアーウォールとの親和性に加えて、ロード・バランサ、SSLセキュリティなどの既存のネットワーク基盤を有効活用できる点も有利だ。

2001年5月に、マイクロソフト社は、IBM社、ヒューレット・パカード社（以下HP社）、コンパック社などと共にSOAPの仕様をW3Cに共同提案しており、本稿執筆時点では、SOAP 1.2がワーキングドラフトの段階にある。恐らく、2001年末までには、ほぼ現状の仕様のままW3Cの勧告となるだろう（もしかすると、SOAPという名称の変更はあるかもしれないが）。もちろん、SOAPだけでWebサービスに必要となるあらゆる通信機能に対応できるわけではないので、SOAPの上位階層で実装される追加機能の標準化活動は今後とも進展していくことになる。また、重要なXMLベースのB2B EC標準であるebXMLにおいても、SOAPをその一階層として使用することが決定されている。ゆえに、SOAPの標準としての将来性はほぼ確実であると言ってよいだろう。少なくとも当面の間は、「Webサービスとは、HTTP上のSOAPを使用して呼び出されるソフトウェア・コンポーネントである」と言ってしまう（技術的には必ずしも正確とは言えないが）現実的に大きな問題は生じないだろう。

その他のWebサービス関連標準

SOAPに加えて、Webサービスの実現に重要となる標準を以下に挙げる。

WSDL

(Web Service Description Language)

Webサービスの入出力インターフェイス仕様を記述するための言語である。分散オブジェクト・コンピューティングの世界で言えば、IDL (Interface Definition Language) に相当する。マイクロソフト社、IBM社、アリバ社の3社により、仕様がW3Cへ提案済みである。

UDDI (Universal Description Discovery and Integration)

企業と企業が提供するWebサービスに関する情報を保管するリポジトリの仕様であり、2000年9月にマイ

クrosoft社、IBM社、アリバ社により提案された。今では、サン・マイクロシステムズ社（以下サン社）、オラクル社、HP社など、他の主流ベンダも参画している。UDDIは電話帳にも、インターネットから必要なサイトを探し出す「Yahoo!」などのサーチ・エンジンにもたとえることができるだろう。分散オブジェクト（コンポーネント）のインターフェイス仕様をIDLで定義し、サーバーのアドレス情報をディレクトリ・サービスやレジストリに保管することで、実行時に呼び出すべきオブジェクトを動的に探し出してアクセスするというORBやRPCの基本的仕組みと同様に、Webサービスのインターフェイス仕様をWSDLで定義し、UDDIレジストリに保管すると考えれば、プログラマにとっては容易に理解できるだろう。

既に、UDDIの仕様に基づいたUDDIビジネス・レジストリが、IBM社、マイクロソフト社、HP社によりホスティングされているが、現在の登録済み参加企業は約280社に過ぎない。また、UDDIは、SOAP、XML Schema、DNSなどの標準に基づいているものの、仕様はベンダグループによる取り決めの段階であり、公的な標準としての地位は得ていない。つまり、将来的にかなりの仕様変更や機能拡張が行なわれることが予測され、UDDIが実際にビジネスの世界で活用できるようになるまでには、かなりの時間を要するであろう。また、競合するebXMLレジストリとの調停も必要となるだろう。しかし、Webサービスが真の価値を発揮するためには、オープンなビジネスのレジストリは必須の要素であり、UDDIは将来的に重要な役割を担うと言える。

また、マイクロソフト社からの提案にすぎない段階であり、現時点では業界の広範なコンセンサスは得られていないものの、今後、重要となるであろう標準としては以下のものが挙げられる。

SOAP-RP (SOAP Routing Protocol)

SOAPメッセージを、ポイント・ツー・ポイントのやり取りをするだけでなく、複数ポイント間で中継していくためのプロトコルである。

DIME (Direct Internet Message Encapsulation)

イメージや音声を効率よくXML形式でパッケージ