

# 第1回 Web サービスの系譜

日本アイ・ビー・エム（株）

丸山 宏 MARUYAMA@jp.ibm.com

日本ユニシス（株）ビジネスイノベーション本部

牧野 友紀 tomonori.makino@unisys.co.jp

日本アイ・ビー・エム（株）

天野 富夫 amano@jp.ibm.com



編集：XML コンソーシアム

## ◇ 連載をはじめるとあって

Web サービスという言葉がIT産業の中で聞くようになって久しいですが、Web サービスの現状はどうなのでしょう？ 本当に使われているのでしょうか？ 世の中がこれで変わっていくのでしょうか？ この12回の連載では、Web サービスにかかわるさまざまなトピックを、毎回単発で読める読み物としてお伝えしようと思います。決して、「Web サービス入門」ではないのですが、一方、`<?xml version="1.0"?>`のようなものが出過ぎないようにもしたいと思います。それでは、スタートしましょう。

## ◇ Web サービスとは？

Web サービスとは、誤解を恐れずに一口でいってしまえばXMLを使った分散プログラミング環境です。よく「Web サービス」という言葉を誤解して、「Web ページを使ったサービス」すなわちオンラインバンキングとかAmazon.comのようなもののことを指して使う人がいますが、それはあやまりです。では、なぜ「Web」サービスなのか、ということが疑問になるかもしれませんが、それは後で述べるように、その成立の経緯に関係があるようです。

「分散プログラミングなら20年以上も前からやっているさ」という方もいるでしょう。特に、19XX年にBerkeleyのSocketライブラリが世の中に出てからは、複数のホスト間でプログラム同士が通信する簡単なプログラムを

書くのはいとも簡単になりましたから、今さら分散プログラミングといわれてもあまりピンとこないかもしれません。

もう少し狭い範囲でいえば、現在「Web サービス」と呼ばれているものは、

- ネットワーク上でサービスとして動いているソフトウェアコンポーネントであり、
  - そのインターフェースはWSDL（Web Services Description Language）で記述されていて、
  - 主にSOAP（Simple Object Access Protocol）というプロトコルで呼び出される、
- というものでしょう。

「ネットワーク上で動いているサービス」という概念も、TCP/IPの初期の頃からあり、DNSやFTPサービスなどは皆さんもいつもお世話になっているものでしょう。一方、「ネットワーク上で動いているサービスを、ソフトウェアのコンポーネントとして考える」というのは、それほど古い概念ではないかもしれませんが、ソフトウェアの部品化と再利用はソフトウェア工学の永遠の課題であり、古くはサブルーチンや抽象データ型、最近ではオブジェクト指向やクラスライブラリなどさまざまな試みがあります。これらの部品が、ソフトウェア開発過程の上でいつ結合されるか、というのは興味深い観点ですが、多くの場合、コンパイル時あるいはリンク時には結合の相手が完全に決まっていなければなりません。一方、ネットワーク上のサービスをソフトウェア部品と考えれば、その部品との結合は実行時に行われる、という考え方もあります（たとえば、HTTPサーバを、Webブラウザから呼び出される再利用可能ソフトウェア部品と考えてみて

ください)。

実はネットワーク上のサービスを使った分散プログラミング環境という概念も今となっては目新しいものではなく、Sun の RPC (Remote Procedure Call, 遠隔手続き呼び出し) から始まって、CORBA やマイクロソフトの DCOM などの分散プログラミング環境がすでに何年もの間、実用に供されています。

それでは、なぜ今また新しい分散プログラミング環境が注目されているのでしょうか？ 表面上は、CORBA との違いは XML を使うかどうかが主な違いみたいに見えます。Web サービスを推進しているベンダは、Web サービスのメリットを業界標準だとか疎結合だとかいろいろ言っています。しかし、使う側にとってのさし当たっての最大のメリットは、マイクロソフトの Windows 系のプラットフォームと、Unix 系のプラットフォームが簡単に統合できるようになったということではないでしょうか？

ただ、Web サービスの元々のコンセプトは、実は分散プログラミング環境にあるのではなく、ビジネスのコンポーネントとして考えるところにあります。この観点については、この連載でまた詳しく述べることにしましょう。

## ◇ Web サービスの系譜

Web サービスという言葉はおそらく、2000 年の 9 月に Ariba、マイクロソフト、IBM の 3 社が Web サービスのレジストリ UDDI (Universal Description, Discovery, and Integration) を発表したときに最初に使ったものだと思います。それに先立つ 1 年くらいの間に、それぞれの会社のソフトウェア戦略家の間で、「Web サービス」という概念が徐々に固まっていったのでしょう。

Web サービスに影響を与えた技術にはいくつかあると思われますが、最も直接的には、HTTP や XML であり、また、その利用領域としては、インターネットを使った e-Business や、企業間のデータ交換である EDI のようなものでしょう。これらの技術や応用の進展に伴って、「今まで多くは個別のシステムだった業務システムが、インターネット技術を通して企業をまたがる統合の対象になったとき、これからのビジネスがどのように変わっていくか」に、主要となる IT ベンダが注目し始めたきわめて自然なことだといえます。

たとえば IBM は 1990 年代に「ビジネスにおけるインターネットの利用」というコンセプトで、e-Business と

いう言葉を使い始めました。この当時はまだ、Web ページで製品紹介をしたり商品の購入を可能にしたりという、システム-ユーザ間の接続が主眼でした。1999 年に、HP 社は「ビジネスがインターネット上のサービスになっていく」というコンセプトで e-Service というスローガンを提唱しました。また、同時にそのころ、ビジネスの分野ではなく、パソコン周辺機器やネットワーク機器の分野で、Sun の JINI や、マイクロソフトの UPnP (Universal Plug and Play) のように、「ネットワークに差し込めば自動的に統合される機器」というコンセプトが登場しつつありました。

「ビジネスがサービスになる」というコンセプトと、「ネットワークにつなげば自動的に統合される」というコンセプトをつなげてみれば、「プラグアンドプレイ e-ビジネス」のようなコンセプトが出てくるのではないのでしょうか？ すなわち、企業の業務システムをインターネットにつなげると、自動的に他の企業群とも接続されて新しいビジネスフローが実現する、というコンセプトです。これこそが、Web サービスの先駆者たちが実現したかった理想だと思います。もちろん、現在の Web サービスの技術はまだまだこの理想に届かない状態ですが、Web サービスの生まれた背景として知っておいてよいと思います。この理想を実現する最初の一步が、非常に多様なプラットフォーム間での相互運用性を保証するための技術の確立であり、そのためには、World Wide Web で広まった HTTP や XML などの技術をベースにすることがベストであるということで、「Web」サービスと呼ばれることになったのだと思われます<sup>★1</sup>。

## ◇ SOAP の泡立ち

Web サービスのアーキテクチャは非常に簡単ですが、2 つの重要な概念があります。それは、送信者と受信者の間で http や ftp などの転送手段に依存しないメッセージ形式でデータを交換する仕組みと、Web サービス利用者が Web サービス提供者と間違いなくデータを送受信できるように、データ転送方法 (送り先アドレスや転送方式) とデータ様式 (データの型や構造と各データ型値の文字表現方式など) を記述する方法の実現です。前者が“封筒と便箋”で例えられるメッセージング・プロ

<sup>★1</sup> CORBA や DCOM、当時の e-Speak 等はそれぞれ独自の技術に基づいていたため、技術上も、ベンダ間の政治的な面でも統合が困難でした。その意味で、あえて「Web」ということで一線を画したかったのだと思います。

トコルの SOAP, 後者がインタフェースの記述言語 (XML 形式) である WSDL により実現されています。

2000 年当初から、オープンソースを管理する Apache から Java で利用する Apache SOAP 1.0 がリリースされ、Visual Basic で利用できる SOAP ToolKit 1.0 がマイクロソフト社より無償で提供されました。これらの開発ツールは、簡単な Web サービスの提供や呼び出しでは、SOAP 規約や XML さえも意識することなく、開発者が慣れ親しんでいるプログラミング言語の中で、通常のアプリケーション・オブジェクトや関数を利用する感覚で扱えるもので、開発者の関心を集め、すぐさま試行的なプログラムの作成が行われました。

なにより開発者が興奮し驚いたことは、(1) 開発ツールの勉強、(2) Web サービスの元になるロジックを記述、(3) Web サービスと WSDL を自動作成し配置、(4) WSDL からプロキシ (代理呼び出し) オブジェクト (関数) を生成し、クライアント・プログラムを作成、(5) インターネット環境で実際に接続、が 1~2 時間足らずでできてしまうことでした。開発ツールは SOAP メッセージの作成や分析処理を提供して通信相手側の実装を隠蔽し、WSDL 中のメッセージ構造の記述をプログラミング言語に対応付けられるため、開発者のプログラム作成作業の省力化が図られているのです。

これまでの、アプリケーション間通信は、データ項目やデータ型が決まっても、通信制御にかかわる部分で、専用線や VAN の施設 (企業間の場合)、通信ミドルウェアの導入、API の理解、アプリケーション・データのバイト数境界または CSV による書式设计、文字コード体系の調整、通信者双方による仕様の理解、ミドルウェア同士の疎通確認、送受信データの整合性確認、等々数週間または月単位のスケジュールが組まれていることが通常ですから、Web サービスの衝撃は相当なものでした。

このような大きな可能性を秘めた Web サービスのビジネスにおける利用は進んでいるのでしょうか? 景気低迷から IT 投資が抑制されていることや、単純な仕組みの裏腹として、インターネット環境で利用することへの不安 (セキュリティ、信頼性)、SOAP メッセージ処理負荷の懸念などから、当初の普及予想から若干遅れています。しかしながら、Amazon の商品検索 Web サービス、Google の Web サービスによる検索インタフェースの提供など、不特定多数の利用者を想定した商用 Web サービスが普及し、企業間ビジネスプロセス連携も、公共性の高い事業を展開する企業を中心に、通信相手のプラットフォーム特性に依存しないことを理由に、Web サービスに

よるインタフェースの提供が増えています。また、2003 年後半から、IT ユーザ企業のシステム要求仕様書に Web サービス利用が明記されることが多くなり、技術としてのコモディティ化が急速に進んでいます。

## ◇ Web サービスは遅いか?

Web サービスを批判する声の中には、「Web サービスは遅くて使い物にならない」というものがあります。確かに SOAP メッセージを使うとデータ転送量は大きくなりますし、また XML 形式を生成したりパースしたりするのもそれなりにオーバーヘッドがあります。現在広く使われている実装は相互運用性に重点が置かれたものであって、高速化の余地がたくさんあるのも事実です。

SOAP メッセージのデータサイズは、もちろん何と比べるかにもよりますが、アプリケーションごとに自分で簡便なデータフォーマットを作るのに比べても、典型的には 2~3 倍くらいになりそうです。CORBA などが比較的小さな LAN 環境で使われるのと想定されるのに対して、Web サービスは、インターネットを介してビジネス間をつなぐことを想定しています。当然、多くのルータや、ファイアウォールなどを通ることになります。典型的な LAN 環境ではネットワークのレイテンシ (伝達時間) が最大数 10ms であるのに対して、インターネット環境では数 100ms と 1 桁高いのが普通です。したがって、Web サービスも、小さなメッセージを頻繁にやりとりすることをあまり想定していません。むしろ、1つのメッセージは、1つのビジネストランザクションに相当するような、大きな粒度になるでしょう。メッセージの粒度が大きくなれば、SOAP のヘッダのオーバーヘッドは相対的に小さくなります。

TCP/IP が設計されたおよそ 4 半世紀前をちょっと考えてみてください。当時は、64Kbps が高速な通信スピードだと思われていた時代です。その時代に設計された TCP/IP は、たとえば telnet で 1 バイトを送るために IP ヘッダ、TCP ヘッダ合わせて 40 バイトのオーバーヘッドを余儀なくされるように設計されています。TCP/IP の柔軟性、堅牢性、相互運用性を犠牲にすれば、きつともっと「効率のよい」プロトコルを設計できたに違いありません。にもかかわらず、皆さんご存知のように TCP/IP は広く使われています。TCP/IP が設計された頃よりもネットワークのスピードもプロセッサのスピードもすでに 1,000 倍以上になっています。Web サービスがその柔軟性、拡張性、相互運用性を提供する限り、パフォーマンスが問題

になることはあまりないといってよいのではないのでしょうか。

### ◇ Does IT Matter ?

2003年5月に、Harvard Business Review に Nicholas Carr という人が「IT Does Not Matter」という8ページの論文を書いて話題になりました<sup>★2</sup>。情報技術が成熟してコモディティ化するにつれ、情報技術をどのように使うかは自明のことになり、企業にとっての競争力の源泉にならない、というのがその骨子です。したがって、情報技術の利用は（電話や電力の利用と同様に）経営課題ではなくなり、現在は企業にとっての戦略的な要職とされているCIO（最高情報責任者）もその役目を終えるだろう、ということです。

確かに、パソコンや、メールや、スプレッドシートのようなオフィスツールはコモディティ化したといえるでしょう。会計業務や、サプライチェーン管理、CRM（顧客管理）などもよいパッケージソフトウェアが出てきて、ITコンサルタント会社に任せれば同業他社に負けないバックオフィスを構築できるでしょう。したがって、このようなIT利用技術のエリアでもコモディティ化が進んでいるといえるかもしれません。

しかし、これらのIT利用技術はまだまだIT利用の入り口に立っているに過ぎない、という考え方もあります。ハーバード・ビジネススクールの Michael Porter という人が書いた「競争戦略論」には、企業の競争力を得る戦略には以下の2つがある、としています。

1. コスト優位戦略－協業他社に比べてより低いコストで製品・サービスを提供する
2. 差別化戦略－協業他社がまねのできない技術やスキルで製品・サービスを提供する

今までほぼ半世紀にわたるIT利用の歴史は、1. のコスト優位戦略に寄与するためのものだったといえるでしょう。企業の既存のビジネスプロセスをIT化することによって、企業のコストを下げたのがその最大の成果だったわけです。

しかし、企業に今までなかった価値を与えるという差別化の観点ではどうでしょうか？ 既存のビジネスプロセスを効率化することだけでは、コスト優位にはなりま

すが、製品・サービスの内容で他社と差別するところまではいきません。情報技術の発展によって、情報の取り扱いに関するコスト構造や、情報の得られるタイミングなどについて、大きな変化が起きています。企業にとっては、その戦略を考えるべき前提に大きな変化があるわけです。この変化は、ゲームでいえばすなわちルールの変化であり、新しいルールに従ってゲームに勝つ戦略を考えるときにきているといえます。新しい戦略には、ビジネスプロセスの見直しが必要で、そのためにビジネスプロセスを柔軟に組み替えることができなければなりません。そのためのWebサービスが大きな役割を果たすと期待されているのです。

情報技術の素晴らしいところは、その利用が発想に応じていくらでも新しい価値を見出せることにあると思います。コスト削減などITの典型的な使い方に関しては、これからもITはどんどんコモディティ化していくでしょう。しかし、ITが可能にする新しい発想は、これからも企業の競争力の源泉になるのではないのでしょうか。

### ◇ おわりに

この号が皆様のお手元に届く8月の終わりには、Web Services Conference 2004 が開催されます。情報処理学会からも基調講演を行います。聴講は無料ですので、Webサービスに興味のある方は参加されてはどうでしょうか？

（平成16年6月1日受付）



★2 2004年5月に、Does IT Matter? というタイトルで本になりました。