

連載 Web サービス

第5回 Web サービスの標準化と相互接続性

日本電気 (株)

藤田 悟 *fujita@cd.jp.nec.com*

富士通 (株)

成田 雅彦 *masahiko.narita@jp.fujitsu.com*

(株) 日立製作所

大場 みち子 *mie_oba@itg.hitachi.co.jp*

日本オラクル (株)

鈴木 俊宏 *Toshihiro.Suzuki@oracle.com*



編集 : XML コンソーシアム

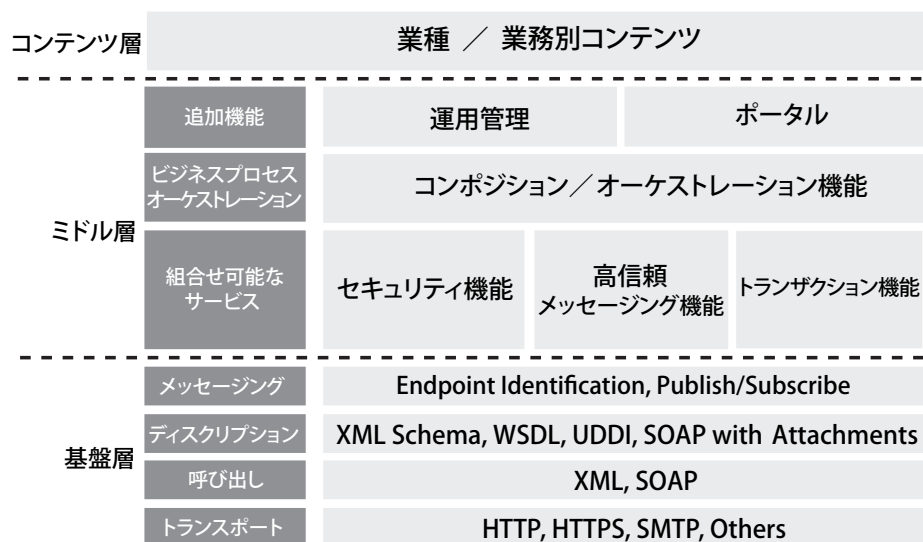
今回は、Web サービスの活用、普及に向けて、1つの大きな課題になっている標準化と相互接続性にフォーカスして解説していきます。

◇ Web サービスの本格的な普及に向けて

Web サービス技術とは、簡単に言うとXMLを利用した分散アプリケーション構築技術です。XMLという機種やプログラム言語に依存しない汎用なデータ表現を用いて、インターネットのような広域に分散したアプリケーション間のデータ交換を行うことを目指しています。しかし、Web サービスの普及が進んでいると言われる米国でも、インターネットでのフロントエンドとバックエンドのアプリケーション連携フレームワークであるEAI (Enterprise Application Integration) における利用が中心です。一方、Web サービスのインターネットでの本格的な利用はまだ始まったばかりです。Web サービスをインターネット上のサービスに適用した事例としては、Google社のインターネット検索サイトやAmazon.com社のオンライン書籍販売などのWeb サービスがあります。日本では、東京三菱銀行のWeb サービスによる輸出入信用状(L/C)取引を始め、大手業界でWeb サービスの活用が始まってきています。

しかしながら、インターネットにおけるWeb サービスの利用が、当初の予想ほどには普及していないというのも事実です。この1つの要因として、SOAPやWSDLなどの基盤層に当たる部分の標準仕様と実際の相互接続性で必要とされる仕様を実現するレベルとの乖離があります。これは製品間の仕様解釈の違いや、実行プラットフォーム、アプリケーション、プログラム言語などの違いにより相互接続性が容易に実現しないことを意味しています。また、基盤層だけでもWeb サービスの構築は可能ですが、さらに複雑な分散アプリケーション同士がWeb サービスを介してより柔軟に連携して動作するためには、高信頼性メッセージ機能、セキュリティ機能、フロー管理機能などのミドル層や、その上でビジネスを表現するコンテンツ層の各層での標準化仕様の開発が重要なファクタとなっています。

そこで、各層におけるプロトコルスタックに対して、ITベンダやベンダ・グループがW3C(World Wide Web Consortium)¹⁾やOASIS (Organization for the Advancement of Structured Information Standards)²⁾等の標準化団体にさまざまな標準仕様案を提案しています。また、標準仕様の作成だけでなく、各層での相互接続性の確保も不可欠です。Web サービスの共通の相互接続仕様の確立を目指して、グローバルに活動する団体としてWS-I (Web Services Interoperability Organization)³⁾があり、国内の活動としては、WS-Iの



Copyright © 2004 by The Web Services-Interoperability Organization (WS-I). All Rights Reserved

図-1 WS-IによるWebサービスの技術スタック

Japan SIG や DOPG などが相互接続性に関する普及活動と、テストの実施などの活動を行っています。図-1に、WS-IによるWebサービスの技術スタックを示します。WS-Iは、このスタックの基盤層から順に、相互接続性の高い仕様の確立を目指しています。

◇ 標準化を推進する W3C と OASIS

Webサービス仕様の標準化はW3CとOASISが業界全体の主導的な役割を担い標準化そのものを推進しています。W3Cは、Webの創始者であるTim Berners-Lee氏が1994年に設立した団体で、MIT、ERCIM、慶應義塾大学SFCがホストを務めています。W3Cの主な作業としてはHTMLからXML、Semantic Web関連に至るまでの標準化であり、基盤ソフトウェアはRoyalty Freeであるべきという方針のもとに活動しています。XML関連の標準仕様ではXML、Xlink、XPath、XQuery、XML Signature、XML Encryption、SOAP、WSDL、XML Schemaといったものがあります。

一方OASISは、主にXMLをベースにしたオープン標準を開発する非営利会員制の標準化団体で、e-ビジネスのオープンな標準の開発、採用を推進することを目的に活動しています。IEC、ISO、ITU、UNECE、NIST、LISA、XBRL、HR-XML、OBG等の団体と協調しながら、XMLのみならずWebサービスのミドル層プロトコルスタックの標準化や、コンテンツ層にあたる業界ソリューションの標準化を行っています。UN/CEFACTとともに推進してきたEDI向けの標準ebXML (Electronic Business using XML)⁴⁾は、業界ソリューションの標準化を代表するものです。

◇ 標準化プロセス、標準仕様の決定方法

具体的な標準化対象技術の話に入る前に、標準化プロセスの進め方について述べておきましょう。標準化団体で、若干の差異はあるものの、新しい標準が提案されると団体内にワーキング・グループが組織されます。そして、仕様策定に参加したい各社が集まって議論を重ねることにより、Working Draftという標準案が作成されます。標準案に至るまでには、メーリングリストでの議論や電話会議が欠かせないものになっているのは、言うまでもありません。その提案された標準案を、たとえば1社1票という投票形式により、オープンな場での「皆が決めた標準仕様」として決定していきます。しかし、最近ではW3CやOASISのような標準化団体に提案をせずに、仕様を一般に公開し、公開した会社主導で、その仕様を普及しようとする動きもあります。この方式では、仕様の提案者だけが仕様の内容について意見交換を行うことができることや、下記に述べるような仕様を利用する時の権利関係が明確にならないことが問題となり、皆が納得する共通仕様に導かれるとは限りません。ちなみに、W3CもOASISも標準仕様の決定に関して、1社1票という投票形式を採用しています。

◇ 標準仕様の権利関係

標準化でもうひとつ大切なことは標準仕様自体の権利関係があります。決定された標準仕様は誰が権利を有するのか、その権利を行使するのかしないのか、どのように

行使するのか、著作権や特許権、知的財産権 (Intellectual Property Right) の問題にも絡みます。そして、標準化団体ごとに採用するライセンス方式が異なるためとても微妙な問題となっています。

インターネット上で利用する Web サービスのようなオープンな標準仕様の多くは、RF (Royalty-Free: ロイヤリティ・フリー) ^{★1} というライセンス方式を採用する方向にあります。しかし、標準化されていない仕様、また一部の標準化団体では RAND (Reasonable And Non-Discriminatory: 妥当かつ非差別的に) ^{★2} というライセンス方式を採用する場合があります。

Web サービスのように、公共性の高い仕様は、標準化団体/業界団体で標準化され、皆が合意した標準仕様として、誰にでも公平にそして自由に安心して利用できるようにする必要があります。特に公的なシステムやソリューションでは公共性を考慮して標準化されたオープンな標準仕様を採用するよう十分に考慮し、また、働きかけていくことがきわめて重要です。

◇ Web サービスの標準化動向

さて、ここからが Web サービスの標準化動向の話に入ります。インターネットの発展に伴い、情報システムも大きく変わってきました。従来は、LAN 環境で特定の相手を対象にシステム構築をしていましたが、インターネット上で、特定できない相手を対象にしたシステムを構築する必要が出てきました。すなわち、イントラネットでは保証されていたセキュリティ、信頼性、相互接続性などをインターネット上で保証しなければならなくなったのです。これは、CORBA (Common Object Request Broker Architecture) ⁵⁾ や J2EE (Java 2 Enterprise Edition) などのイントラネット環境にふさわしい分散アプリケーションの構築技術をインターネットに対応させるために、いわば、「再構築」することを意味します。

図-2 に、RosettaNet ⁶⁾ や ebXML、CORBA との比較を示します。RosettaNet や ebXML は、XML ベースの EDI システム構築への適用でした。RosettaNet はもともと高信頼メッセージングなどのミドル層の機能を持っていましたが、最近では標準の ebXML や Web サービスなどを利用する方向に移りつつあります。また、ebXML では、基盤層に SOAP を採

^{★1} 誰にでも公平に無料でそのライセンスの利用を認めます。

^{★2} 妥当かつ非差別的に、つまり、誰に対しても分け隔てなくライセンスを供与するが、ライセンス料は徴収する可能性があります。

用し、その上のミドル層として高信頼メッセージングの仕様やレポジトリの仕様を作り、さらに、コンテンツ層として、ビジネスのための共通コアコンポーネントの策定まで、一貫した EDI 標準体系を築いています。ebXML は、日本やアジア、ヨーロッパに普及を始めています。すでに、電子商取引推進協議会や ebXML アジア委員会などが ebXML の相互接続テストを実施して、ebXML 製品間の相互運用性の充実をはかっています。

Web サービス本体の話に戻しましょう。分散アプリケーション基盤のための再構築の対象となる技術は、①基盤層 (SOAP 等の下位プロトコルの基本通信層)、②高信頼メッセージング (到達保証を行う非同期通信やルーティングなどの拡張機能層)、③セキュリティ (認証、プライバシー、ライセンス、電子署名)、④トランザクション、⑤ネゴシエーション (2 つのシステム間で通信・上位コミュニケーションスタックの交渉)、⑥フロー・ビジネスプロセス定義など、広範囲に及んでいます。それぞれについて、以下に順に説明します。

(1) 基盤層

Web サービスの基盤層の機能は、以下のとおりです。

- ① SOAP は、XML データを通信する方式です。下位のメッセージトランスポートとしては、主に HTTP を用います。
- ② WSDL は、Web サービスの通信プロトコルを XML ベースで記述するためのインタフェース記述言語です。通信メッセージの引数や戻り値のデータ型などを中心に定義します。
- ③ UDDI は、Web サービスの各種の情報 (企業情報、フロー情報等) を格納し、引き出すための機能を提供します。SOAP と WSDL については、W3C が標準化を進めており、UDDI については、OASIS が標準化を進めています。

(2) 高信頼メッセージング

インターネットの通信は、安定性や確実性に問題があります。そこで、上記の基本通信機能に加えて、メッセージングに高信頼性を付加する機能が必要になります。たとえば、送達保証を行う非同期通信は、サービス連携を考える上で重要な要素になります。主な拡張機能は以下のとおりです。

① 送達保証

通信中にエラーが発生した場合でも、メッセージの自動再送を行うことによって、メッセージを確実に相手に送付します。

② 重複受信防止

同じメッセージを重複して受信しないようにします。メッ

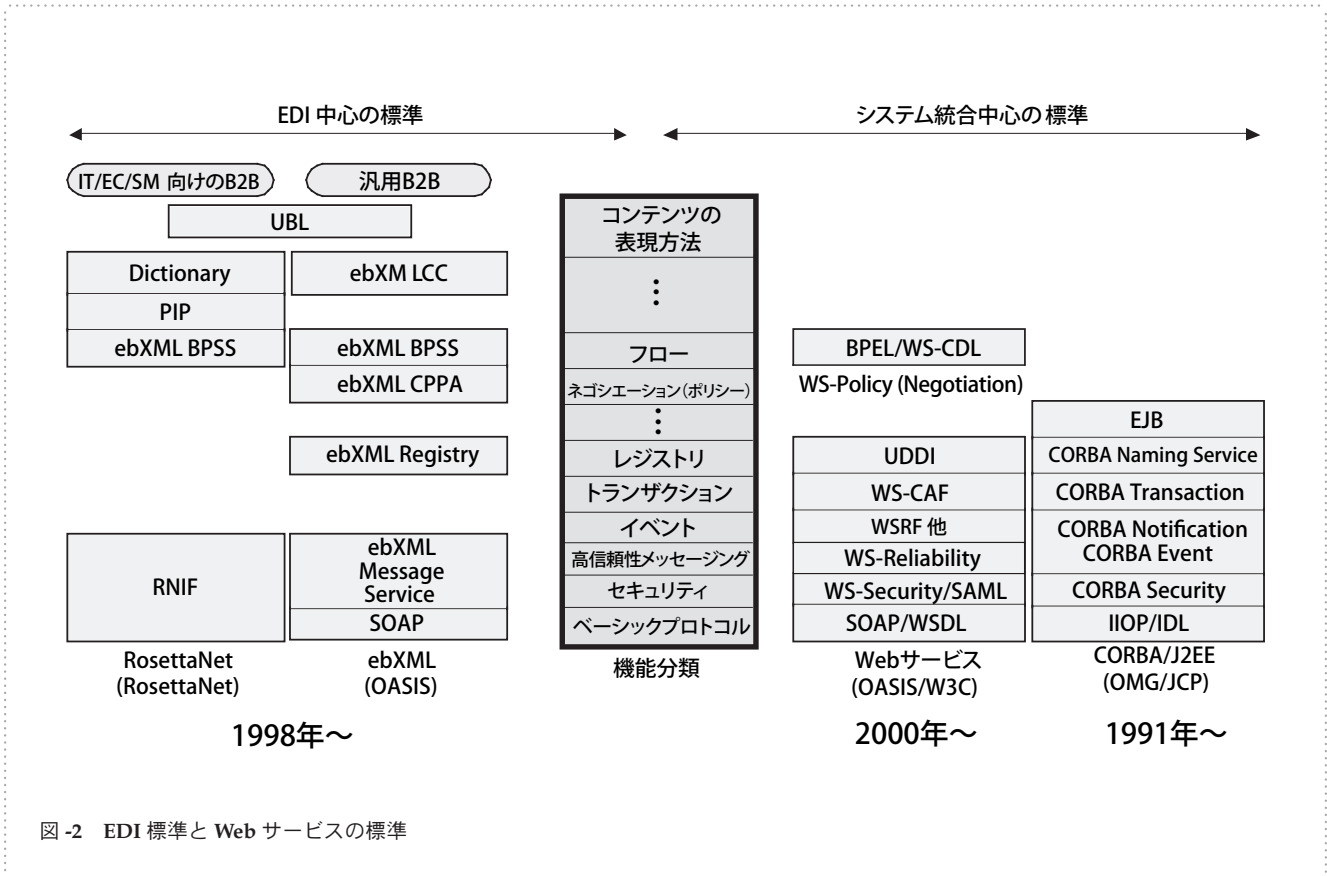


図-2 EDI標準とWebサービスの標準

メッセージごとに ID を付けて受信側がメッセージの同一性を確認します。

③ 順序保証

メッセージにシーケンス番号をつけて、送信側がメッセージを送った順番に受信側で処理できることを保証します。高信頼メッセージングの仕様には、OASIS で標準化中の WS-Reliability のほか、標準化団体に提案されていない WS-Reliable Messaging など、複数の仕様が存在しています。

(3) セキュリティ

Web サービスにおいて、安全なメッセージ交換の仕組みを提供する機能です。主な機能は以下のとおりです。

- ① SOAP メッセージの完全性(Integrity) や秘匿(Confidentiality) の保証
- ② セキュリティトークン (認証トークンやアクセス権限トークン) を作成するための仕組み
- ③ セキュリティトークン、セキュリティコンテキストやポリシーを送受信するためのプロトコル
- ④ 署名や暗号化で必要となるセッション鍵を作成するためのセキュリティコンテキストの表現方法
- ⑤ セキュリティコンテキストからのセッション鍵の取り出し方法

セキュリティ機能の仕様には、OASIS で標準化中の WS-Security や SAML のほか、標準化団体に提案されていない WS-Trust や WS-SecureConversation などがあります。

(4) トランザクション

Web サービス間のトランザクションは、分散環境において処理の一貫性や信頼性のある操作を可能にするための仕組みです。主な機能は以下のとおりです。

① Atomic Transaction

短期間のトランザクションに用いられ、データ更新の Atomic 性を実現します。

② Business Activity

長期間のトランザクション用で、ビジネスレベルの例外処理などに適用します。トランザクション中にリソースをロックせず、操作結果はすぐにリソースに反映して永続化し、例外がおきた時に補償処理を行います。たとえば、ホテルと飛行機の予約を同時に行う時に、このトランザクションを用います。

現在は、Business Activity に相当するトランザクション処理は、各アプリケーションが個別に業務レベルの実装として実現しています。したがって、こうした Web サービス共

通のトランザクション機能が受け入れられるには、まだ、時間がかかりそうです。

トランザクション機能の仕様には、OASIS に提案中の WS-CAF のほかに、標準化団体に提案されていない WS-AtomicTransaction や WS-BusinessActivity, WS-Coordination などがあります。

(5) ネゴシエーション (ポリシー)

Web サービス間でネゴシエーションを行うためにリモートからポリシー情報にアクセスする仕組みです。その主な機能は以下のとおりです。

- ① 認証方式、プライバシー、QoS 等の Web サービスのポリシー情報を記述するためのフレームワーク
- ② ポリシーに関する WSDL や UDDI の修正
- ③ 利用可能な言語、仕様のバージョン等の基本的な Policy Assertion を定義する言語

ネゴシエーション機能に関連する仕様には、まだ標準化されていない WS-Policy, WS-PolicyAttachment, WS-PolicyAssertion などがあります。

(6) フロー (ビジネスプロセス)

フローとは、企業間でビジネスを行う際のメッセージ交換に対するプロセスを定義して実行する機能です。この機能によって、相手から受け取ったメッセージに対して、どのようなメッセージを返せばよいのかななどを明示的に定義することができます。

フローに関連する仕様には、OASIS で標準化中の BPEL4WS, W3C で標準化中の WS-CDL などがあります。

◇ 相互接続性の確立に向けて

Web サービスの大きな魅力は、異なるプラットフォーム間の相互接続性の高さです。しかし、初期の Web サービスは、各プラットフォーム間に非互換性が残り、相互接続できないケースがありました。そもそも、Web サービスの中で使われる XML は、SGML を基にして拡張されたものですが、インターネットの表現言語として広く利用されている HTML と同様に、データ構造をタグの階層として表現でき、人が読めるデータ表現であることが特徴です。この特徴が災いして、「XML なら、簡単に作れるし、簡単に繋がる」という安易な期待があったのかもしれませんが、実際に、Web サービスの非互換性の主たる原因は、SOAP と WSDL の仕様や、パケットの発行のタイミングを記述したプロトコルの仕様そのものにありました。仕様書の中に誤っ

た記述や、矛盾する表現があり、仕様を実装する人に混乱を与えました。また、これらの仕様は、今日の XML を支える技術となっている XML Schema の仕様が確定する前に策定されたということも、仕様内容が混乱する大きな要因であったと思います。

SOAP と WSDL に関して、少し詳細に見てみましょう。たとえば、SOAP 1.1 の仕様書には、“encodingStyle” というデータのエンコーディング方法を指定する属性があります。ここに既定値を指定すると、SOAP 1.1 の仕様書に規定されたエンコーディングを採用することになります。SOAP 1.1 の仕様書に示されるエンコーディングには、データ型の表記法が規定されており、特に配列に関しては、SOAP 1.1 に特有の表現方法が記されています。たとえば、次のような配列要素の表現方法が記載されています。

```
<myFavoriteNumbers
  SOAP-ENC : arrayType="xsd:int[2]">
  <number> 3 </number>
  <number> 4 </number>
</myFavoriteNumbers>
```

この配列の表記法では、SOAP-ENC : Array という特殊なデータ型を利用しており、また、arrayType に“xsd:int[2]”という形式で、配列の要素型と要素数を定義するフォーマットを利用しています。これは SOAP 1.1 独特の型定義形式であり、XML Schema による検証を難しくしていました。そこで、この SOAP 1.1 のエンコーディングを利用しないで、XML Schema で規定している方法として、要素の繰り返しによって配列を表現する方法も試みられ、これが製品間の実装の差となっていました。

また、SOAP 1.1 や WSDL 1.1 の仕様書に現れる XML の例には、多くの誤りがあります。仕様書の本文中の説明で示された構文と、XML Schema で規定された構文に矛盾がある場合もあります。このような誤りの多さについても、XML Schema が未完成の時期に仕様を作った影響であり、現在であれば、XML Schema に従った検証を行うことによって、XML 事例の誤りは少なくおさえられたでしょう。最近では XML Schema も完成しており、すべての Web サービスが XML Schema を利用することによって、データ型に唯一のスキーマが存在し、首尾一貫した、より相互運用可能な Web サービスの構築が期待されます。

◇ WS-I による相互接続のガイドライン作り

このような SOAP 1.1 や WSDL 1.1 に対する仕様の乱れを正しく修正して、Web サービスの本来の性質である相互接続性を確保するための活動が WS-I で行われています。WS-I では、SOAP 1.1, WSDL 1.1, UDDI 2.0, XML 1.0, XML Schema Part1 Part2, HTTP 1.1 などの仕様をベースにした互換性の高い Web サービスを実現するためのガイドラインとして、Basic Profile を策定して、公開しています。2004 年 8 月には、Basic Profile 1.1 が公開され、MIME を利用した attachments 付きの Web サービスについても、相互接続ガイドラインが示されました。

Basic Profile では、各仕様書に現れる矛盾や誤りを訂正するだけでなく、仕様書間に現れる重複した記述や、相互に矛盾する可能性のある記述に関する留意点などをまとめて、相互接続できる Web サービスのための共通仕様作りを目指しています。この Basic Profile の策定に当たって重視されたのが XML Schema でした。問題点が多かった encodingStyle の使用を禁止して、その代わりに、XML Schema をベースにして、SOAP 1.1 や WSDL 1.1 の仕様を解釈しなおす作業を行ったのです。今では、多くの Web サービス製品は、WS-I のガイドラインに沿った Web サービスを実現するためのモードをサポートしたり、さらには、互換性の検証を行うツールを充実したりしています。

Basic Profile 1.1 が規定しているのは、SOAP, WSDL, UDDI レベルの相互接続性です。WS-I では、WS-Security に基づく Security Profile についてもガイドラインの作成を進めており、2004 年 5 月にドラフト版を公開しました。今後も、基盤レベルの相互接続性から、Web サービスの拡張仕様に向かって、高い接続性を持つ Web サービスのガイドライン作りが進められていくことになります。

Web サービスの仕様だけでなく、その上に構築されるビジネスレベルの仕様についても、相互接続性を確保するための努力が重要です。すでに述べた ebXML のメッセージングの仕様も、V1.0 から V2.0 に移る段階で、スキーマが整備されたことで、互換性検証がやりやすくなりました。実際に、複数のベンダが集まった相互接続試験では、問題が生じた時に原典とすべきものが明確であり、スムーズに試験を進行させることができました。また、現在、業界ごとに、Web サービスのコンテンツ層であるビジネスレベルの XML データの標準仕様の策定が進められていますが、XML のタグ名の決定だけに終わらず、XML Schema を用いて、データ構造を定める努力をすることで、今後の相互接続性の高い Web サービスの実現につながっていくと考えます。

◇ おわりに

これまで Web サービス技術の標準化と相互接続の動向を駆け足で説明してきました。Web サービス技術は、インターネットに適した分散アプリケーションの構築技術として、基盤層、ミドル層、コンテンツ層というプロトコルスタックにわたって、標準仕様が再構築されつつあります。また、「XML は簡単」という落とし穴が、標準の仕様までも不正確なものにしてしまい、非互換性の要因の 1 つになっていたことも説明しました。最近では SOA (Services Oriented Architecture) と Web サービス技術を混同している人が多いことも、Web サービスに対しての十分な理解を阻害する原因にもなっています。SOA は考え方であり、Web サービス技術はプロトコルの技術スタックです。

今では XML 関連の標準技術も整備され、WS-I の相互接続のガイドラインも加わり、本来求めていた相互接続性の高い Web サービス技術が現実のものになってきました。今後も、Web サービスが皆の共通の資産となるよう、期待していきたいと考えています。

参考 URL

- 1) W3C : <http://www.w3.org/>
- 2) OASIS : <http://www.oasis-open.org/>
- 3) WS-I : <http://www.ws-i.org/>
- 4) ebXML : <http://www.ebxml.org/>
- 5) CORBA : http://www.omg.org/technology/documents/corba_spec_catalog.htm
- 6) RosettaNet : <http://www.rosettanet.org/>

(平成 16 年 10 月 12 日受付)

