

# ソフトウェアサービス技術への いざない

青山 幹雄

南山大学 mikio.aoyama@nifty.com

## ソフトウェアサービス技術とは

### ■e-ビジネスを実現するソフトウェアサービス技術

電子商取引を始めとする企業全体の活動をインターネット上で行うe-ビジネスの実現が求められている<sup>3), 7)</sup>。e-ビジネスの基盤は、インターネットの非集中アーキテクチャと「どこにも繋がる」ユービキタス性にある。したがって、e-ビジネスを実現するソフトウェアにはプラットフォームや企業間にまたがる相互運用性、要求や環境の変化に応じた俊敏な進化、などの新たな要求がある。従来のクライアント・サーバアーキテクチャに基づく密結合の分散オブジェクト環境ではこのような要求を十分に満たせないことから、新たなソフトウェアアーキテクチャ、ソフトウェア開発技術が求められてきた。

これに対し、XMLの記述性と相互運用性に着目し、XMLを用いてプログラムのインタフェースを記述し、インターネット上でプラットフォームによらず相互運用を可能とする技術が開発された。このように、業務や機能単位でプログラムのインタフェースを隠蔽し、その提供機能を抽象的に記述したものを「サービス」と呼ぶ。サービスはコンポーネント<sup>1)</sup>をインターネット上で遠隔に再利用できるように進化させたものである。

サービスはインターネット上ですでに広く提供されている。たとえば、図-1にインターネット上での物品購入を簡略化して示す。「物品購入」は、「物品検索」「価格交渉」などのプロセスからなるが、これらのプロセスは、「物品検索サービス」などのWebページを利用して実行できる。

さらに、このようなサービスをコンピュータ同士が直接連携して、「物品購入」の例のような、より高度な

サービスを実現することが期待される。このようなサービスを実現する技術を本稿ではソフトウェアサービス技術と呼ぶ<sup>2), 4)</sup>。

このような考えに立つと、e-ビジネスはさまざまな業務をサービスとして連携することにより実現される。したがって、ソフトウェアサービス技術はe-ビジネスを実現する技術といえる。

ソフトウェアサービス技術の中で、サービスを実現する基盤技術の1つの体系として2000年から急速に進化しつつある技術がWebサービスと呼ばれている<sup>☆1</sup>。さらに、サービスを開発するためのさまざまな技術が活発に研究・開発されている。このようなサービス中心に考えシステムを実現する技術をサービス指向(Service Oriented)とも呼ぶ。

また、Webサービスは、主としてB2CやB2Bを実現するサーバ主導の技術として発展してきた。これに対し、

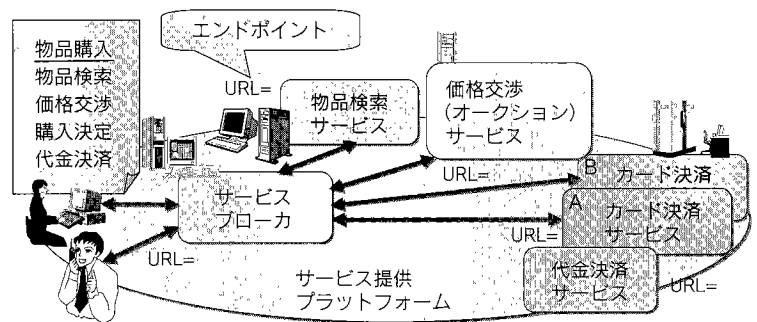


図-1 ソフトウェアサービス技術とは

☆1 従来のHTMLで記述したWebページのWebサービスと区別するためXML Webサービスとも呼ぶが、本稿では、単に、Webサービスで統一する。

P2P (ピア・ツー・ピア)<sup>15)</sup>のサービスをピアサービスと呼ぶ。

本稿では、Webサービスを中心とし、ソフトウェアサービスの基盤技術と開発技術の両面から、その全体像を紹介する。

### ■プログラムからサービスへの進化

「サービス」という言葉は、オブジェクトやコンポーネントと同様、一般によく使う言葉であるので、これまでの技術と比較し、位置付けを明らかにしておく必要がある。図-2にオブジェクトからサービスへの技術の進化を示す。

オブジェクトは、実世界をモデル化するデータをその処理手続きによってカプセル化したものである。コンポーネントは、1つあるいは複数のオブジェクトをまとめて独立に扱えるようにパッケージ化したものである。代表例として、WindowsのCOMコンポーネント、JavaのJavaBeansやEJB (Enterprise Java Beans) がある。これらのコンポーネントは、その実体を組み込んで利用するので密結合となり、かつ、コンポーネントを所有することになる。

一方、インターネット上でソフトウェアをレンタル利用するASP (Application Service Provider) が出現した。ASPではERP (Enterprise Resource Planning) などの高価なパッケージソフトウェアの機能を遠隔利用できる。このような、遠隔利用モデルを拡張すればコンポーネントやパッケージソフトウェアの機能をインターネット上で遠隔に利用し、かつ、組み合わせることができる。その結果、コンポーネントを所有することなく、柔軟で高度な機能を利用できるようになる。

もう1つの流れは、e-マーケットプレイスやSCM (Supply Chain Management) など、インターネット上

で企業や業種を超えて商取引を行うアプリケーションが要求されていることにある。レガシーEDI (Electronic Data Interchange) と呼ばれる従来のEDIでは、個別に取引手順を定め、それがアプリケーションに組み込まれているため、このような要求に対応できない。

このような流れに沿って、ソフトウェアサービス技術は次のような新たな要求を満たすことを目的としている。

- 1) プラットフォーム独立なソフトウェア再利用: プログラミングインタフェースを隠蔽し、プラットフォーム、言語、オブジェクト/コンポーネントモデルから独立してソフトウェアをサービスとして再利用。
- 2) 疎結合動的ソフトウェアアーキテクチャ: 実行時に必要なサービスを動的に探索し、結合することによる非集中的で疎結合な動的ソフトウェアアーキテクチャの実現。
- 3) 組織間相互運用性: B2Bなどを実現する、ファイアウォールを超えた組織間の相互運用性の実現。

このようなサービス技術を捉えるため、図-3に示す3階層参照モデルを筆者らは提案している。サービス空間は、上位の多様なビジネスやユーザの要求をサービ

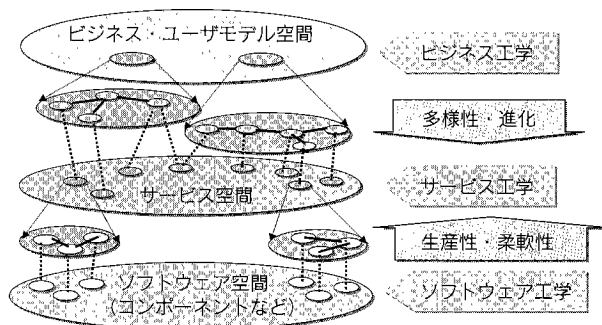


図-3 ビジネス・サービス・ソフトウェア

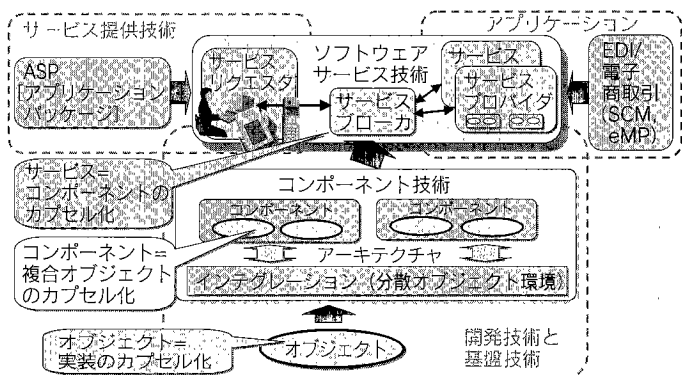


図-2 オブジェクトからサービスへ

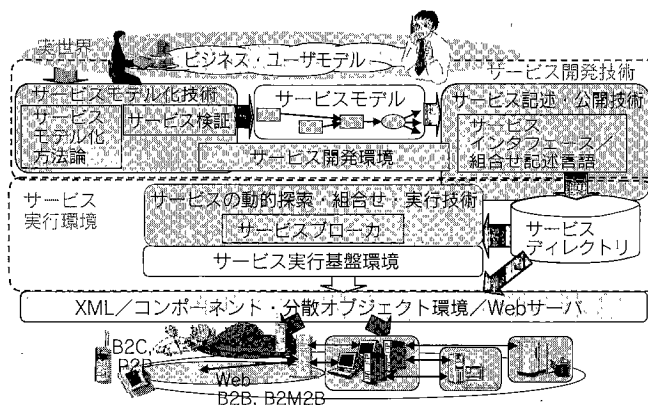


図-4 ソフトウェアサービス技術の枠組み

スの組合せで実現する。一方、ソフトウェアの実装技術を隠蔽し、抽象的で相互運用性の高いモデルを提供する。

### ソフトウェアサービス技術の枠組み

ソフトウェアサービス技術の枠組みを図-4に示す。ソフトウェア開発と同様、サービス開発技術とサービス実行環境の2つに分けることができる。サービス実行環境はWebサービスを中心に技術開発、応用が進んでいる。サービス開発技術は種々の試みはあるが、確立されているとは言い難い。

以下、サービス実行環境と開発技術それぞれの全体像を示す。

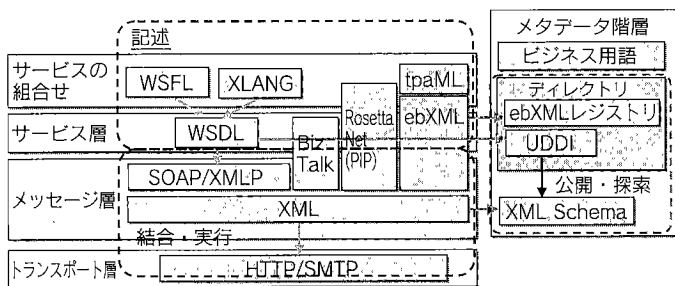


図-5 サービス実行環境のフレームワーク

技術	内容	開発	標準化
SOAP/XMLP	XMLベースの軽量手続き呼出しプロトコル	Microsoftほか	W3C:V1.1 (2000年5月), V1.2WD (2001年7月)
WSDL	サービスインタフェース定義	IBM, Microsoft (2000年9月)	W3C V1.1(2001年3月)
WSFL	ワークフローに基づくサービス連携	IBM (2001年5月)	-
XLANG	ビジネスプロセス記述のためのWSDLの拡張	Microsoft (2001年5月)	-
UDDI	サービスディレクトリ(サービスの属性とそのエンドポイントの登録)	UDDI Initiative (Ariba, IBM, Microsoft他) V1 (2000年9月) V2Draft (2001年6月) 正式サイト運営 (2001年5月)	
ebXML	XMLベースのEDI標準	UN/CEFACT, OASIS仕様公開 (2001年5月)	
tpaML	電子商取引のプロトコルと契約文書等の標準化	IBM	ebXMLへ提案
RosettaNet	SCMのビジネスプロセス定義と実行	RosettaNet (Intel他電子部品業界から)	

表-1 サービス実行環境の主要要素技術

### ソフトウェアサービス実行環境

#### ■サービス実行環境の技術フレームワーク

サービス実行環境の要素技術を階層化して、図-5に示す。また、各要素技術の概要を表-1に示す。この中で、SOAP, UDDI/WSDL, ebXML, RosettaNetの詳細は、それぞれの解説を参照願いたい。また、この他のB2Bのフレームワークは文献18)にサーベイがある。

#### ■サービス指向アーキテクチャ

図-6にサービス指向アーキテクチャSOA (Service-Oriented Architecture) とその実行プロセスを示す。サービス実行環境では、疎結合で動的なサービスの発見、結合、実行を実現するためのソフトウェアアーキテクチャとして、ブローカーアーキテクチャをとる。サービスの利用者をサービスリクエスタ、サービスの提供者をサービスプロバイダと呼ぶ。サービスブローカーはサービスリクエスタの要求に応じて、適切なサービスを探索し、起動する仲介を行う。

サービスブローカーは、単に、ディレクトリを探索してサービスの発見をするだけでなく、付加価値を持つ高度な仲介機能も求められる。たとえば、サービスの組合せやルーティング、QoS (Quality of Service) や品質保証、セキュリティなどの非機能的な要求もある。

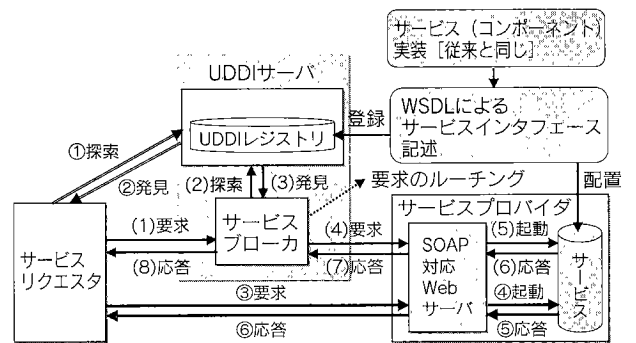


図-6 サービス指向アーキテクチャ

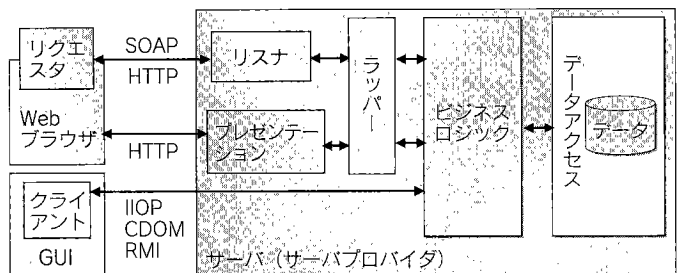


図-7 サービス指向アーキテクチャと既存アーキテクチャ

サービスプロカーはサービス提供の鍵となる技術である。

図-7にサービス指向アーキテクチャを従来のWebアプリケーション、分散オブジェクト環境のアーキテクチャをあわせて示す。サービスリクエストはWebブラウザの場合と他のサーバの場合がある。HTTP、SOAPという標準プロトコルによって、プラットフォームや言語によらない相互運用性の実現できる点で分散オブジェクト環境と異なる。

## ■ サービスの記述と探索・実行

### [サービスの記述]

WSDL (Web Services Description Language) はサービスのインタフェースをXMLで記述する言語である<sup>8)</sup>。分散オブジェクト環境のインタフェース定義言語に相当する。図-8にWSDLの構造を示す。右端のプログラムの機能をサービスとして提供するためのラッピングを行う。バインディングはリクエストの呼び出しプロトコルへのマッピングを行う。SOAP (Simple Object Access Protocol) はXMLベースの軽量手続き呼び出しプロトコルである<sup>9, 6)</sup>。左端のリクエストからみると、サービスはURLで表されるエンドポイント (endpoint)、すなわちポートの集合となる。

インタフェースが相互運用可能であるためには、語彙とそのスコープをリクエストとプロバイダで共有する必要がある。そのため、XML名前空間 (XML name space) を用いる。

### [サービスの公開・探索]

サービスを公開し、検索するためのディレクトリがUDDI (Universal Description, Discovery and Integration) である<sup>20)</sup>。インターネット上でビジネスやサービスの電話帳になる。その構造も表-2に示すように、電話

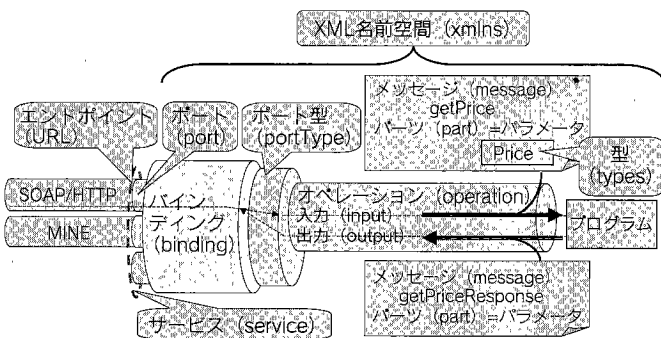


図-8 WSDLの構造

<sup>☆2</sup> SOAPの高度化としてXMLP (XML Protocol) が検討されている。

帳に似せてある。ここで、グリーンページが従来にない新しい概念で、コンピュータ間でサービスを結合するための情報を提供する。

また、ビジネス、サービスを検索するために、categoryBagに分類情報を登録できる。ここには、業務 (ビジネスプロセス)、業種分類 (NAICS)、製品分類 (UNSPSC)、地理的位置 (ISO3166) を分類キーとして保持できる。

UDDIはビジネスとサービスに関する情報を提供するディレクトリであるから、技術面だけでなくビジネス面でも重要である。現在、IBMとMicrosoftによりグローバルなUDDIが運用されている<sup>20)</sup>。今後、企業内や業界内など、さまざまな形態のUDDIの運用も考えられる。

### [サービスの組合せ]

Webサービスをコンポーネントとして組み合わせて、ビジネスプロセス全体にわたるサービスやより高度なサービスの提供が求められる。特に、UDDIにより実行時にサービスを探検し、動的なサービスの組合せが可能となる。

このようなサービスの組合せを記述する言語として、WSFL (Web Service Flow Language)<sup>13)</sup> やXLANG<sup>19)</sup> が提案されている。

## ■ ebXMLとRosettaNet

ebXML (e-business XML)<sup>11)</sup> やRosettaNet<sup>16)</sup> は、Webサービスとは独立に、実際の電子商取引のための標準として開発されてきた。しかし、その技術はWeb

分類	目的	内容
ホワイトページ (White Page)	ビジネスの公開と検索	ビジネス情報 (businessEntity): 個々のサービスプロバイダ (企業, 組織) の属性 (名前, 住所, 電話番号, URL 業種など) とその提供するサービスへの参照など。複数のサービス情報への参照を保持。
イエローページ (Yellow Page)	サービスの公開と検索	サービス情報 (businessService): 個々のサービスの属性とバインド情報への参照など。サービスはUID (Universal Unique Identifier) で一意に決まるserviceKeyで定義。複数のバインド情報への参照を保持。
グリーンページ (Green Page)	サービスの結合 (bind)	バインド情報 (bindingTemplate): 個々のサービスを結合する方法とそのアクセスポイント (URLなど) を規定し、サービス型 (tModel) への参照を保持。 サービス型 (tModel): サービス群を分類して型付けしたもので、複数のサービスから参照される。

表-2 UDDI V1のデータ構造

サービス技術と共通点が多い。

ebXMLはXMLを用いて、インターネット上で相互運用可能なEDIの標準として開発された。さらに、取引者間での取引手続きと交換文書の標準としてtpaML (trading-partner agreement Markup Language) が提案されている<sup>9)</sup>。

また、RosettaNetは、パソコンなどの電子部品業界が主導して、SCMなどの構築を目的とした標準である。

最近、ebXMLやRosettaNetも、実行プロトコルとしてSOAPを用いることが表明されている。

### ソフトウェアサービスの開発技術

#### ■サービス開発技術のフレームワーク

ソフトウェアサービス技術は、オブジェクト指向技術やコンポーネント技術を基礎とするので、サービス開発技術もオブジェクト指向開発技術が基礎となる。たとえば、サービス実行環境の仕様もオブジェクト指向の標準表記法であるUML (Unified Modeling Language) で記述されている。

しかし、サービス開発では、動的なサービスのモデリングや制御において従来にない新たな課題も提起されている。このような、動的なサービスのモデリング、検証技術の開発が求められている。

#### ■サービス開発方法論

本稿では、サービス開発技術を図-9に示す2つの視点に分けて紹介する。

##### [ビジネスモデル駆動型]

ソフトウェアサービス技術の主目的の1つはB2Bの電子商取引や電子政府などの組織間にまたがる連携の実現にある。このような分野では、従来ビジネスモデル毎に個々に開発してきたソフトウェアを、サービスを組み合わせて効率的に提供することが期待できる。

たとえば、図-10はオークションなどの異なるビジネスモデル<sup>12), 17)</sup>の構造を抽出したものである。これらは、構造的にはハブあるいはブローカーアーキテクチャである。したがって、サービス指向アーキテクチャと親和性が高い。

図-11は簡略化したブローカーの構造を表すクラス図である<sup>5)</sup>。RFBはオークション方法に基づき、応札した売り手からある選好に基づいて購入する製品を決定する。その結果は、取引条件とともに、取引として保持される。

さらに、動的な振舞いの視点からみると、取引の役割の動的な変更やスコープの制御、ネゴシエーション制御など、ブローカーの設計では複雑な振舞いの制御が求められる。

##### [ユーザモデル駆動型]

ユーザモデル駆動型はB2Cの電子商取引などにおいてユーザの振舞いを中心に設計する方法である。これには、次の2つのアプローチがある。

- 1) ユーザとサービスの相互作用モデル: ユーザがサービ

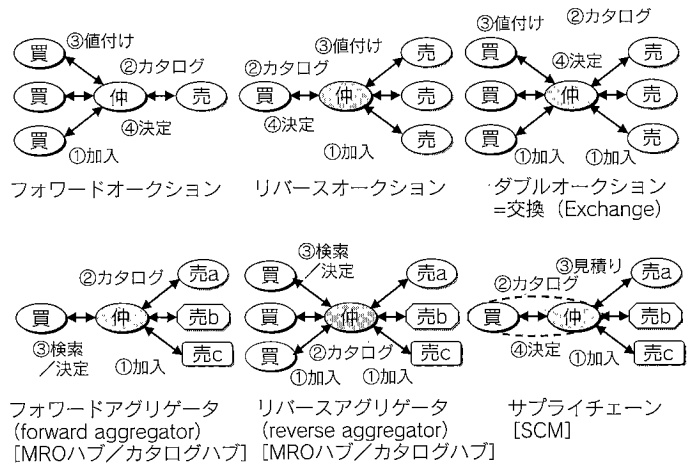


図-10 ビジネスアーキテクチャパターン

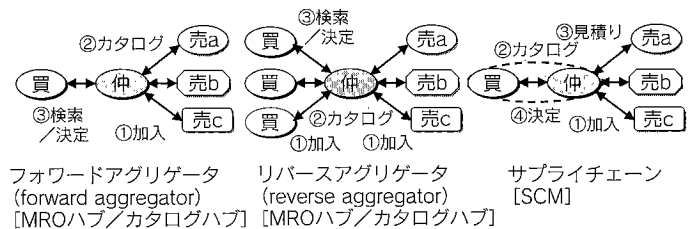


図-11 簡略化したブローカーのクラス図

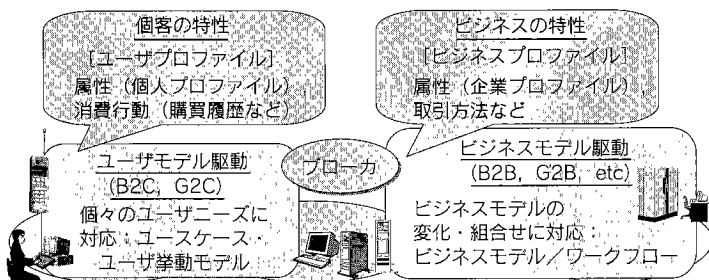


図-9 サービス開発技術の枠組み

スを利用する手順をUMLのユースケースとそのシナリオなどで記述する。

- 2) ユーザの状態遷移モデル: サービスの構造をグラフでモデル化し、グラフ上でのユーザの状態遷移をモデル化する<sup>14)</sup>。

今後、サービスの抽象度の高さに焦点を当て、e-ビジネスを効率よく実現する開発方法論が求められる。

## ■ サービス検証技術

電子商取引のようなネットワーク上で多くのシステムが関与するシステムの信頼性、品質の確保は一層困難になると予想される。

このため、モデル検証 (model checking) 技術を応用し、モデルや設計段階でその構造的な特性を検証する試みが行われている<sup>21)</sup>。

## ソフトウェアサービスコンポーネントの提供

サービスをコンポーネントとして提供するには、次の2つのアプローチがある。

- 1) インターネットの特性を活かした、新サービスの提供: インターネット上でどこからでも1度ログインすればサービスが受けられるワンストップサービスのためのユーザプロフィール情報サービスなどの提供が予定されている。たとえば、MicrosoftのHailStormと呼ばれるサービス群がある。
- 2) 既存ソフトウェアのサービス化: 図-7に示したように、既存のコンポーネントやパッケージをラッピングし、サービス化して再利用する。

## 今後の課題

ソフトウェアサービス技術は、まだ、黎明期にある。実行環境・開発技術、そして、その利用技術・ビジネスの両面で研究・開発と実践経験の蓄積が必要である。また、より上位のビジネスの構造やモデル化の研究も必要である<sup>10)</sup>。

## ま と め

ソフトウェアサービス技術はXMLによってコンポーネント技術が進化した技術である。その意味で、従来

技術の延長線上にある。しかし、分散オブジェクト環境で実現できなかった異種プラットフォーム間の相互運用を実現するなどのインターネット上の本質的な問題の解決を目指している点で重要である。また、サービスのディレクトリなど、技術とビジネスの両面で今後重要な役割を果たすと期待される技術も新たに開発されている。

このような背景から、筆者らは、2000年度から「動的ソフトウェアサービスに関する調査研究」を行っている<sup>4)</sup>。また、ソフトウェアサービス技術に関する国際ワークショップWebSE (Web Service Engineering) 2002の開催も企画している<sup>22)</sup>。今後、我が国の技術者・研究者が本分野へ参画されることを期待する。

最後に、本稿をまとめるにあたりご討論いただいた「動的ソフトウェアサービス技術調査研究」WGの委員各位、ならびに、日本アイ・ビー・エムの丸山宏氏、マイクロソフトの萩原正義氏に感謝する。

## 参考文献

- 1) 青山幹雄: コンポーネントウェア: 部品組立て型ソフトウェア開発技術, 情報処理, Vol.37, No.1, pp.71-79 (Jan. 1996).
- 2) Aoyama, M.: Intelligent Software Services over the Internet, Kangassalo, H. et al. (eds.), Information Modeling and Knowledge Bases, IX, IOS Press, pp.128-135 (2000).
- 3) 青山幹雄: コンピュータが消える日: インターネット時代のソフトウェア, 情報処理, Vol.41, No.5, pp.523-527 (May 2000).
- 4) 青山幹雄ほか: 動的ソフトウェアサービス技術に関する調査研究WG, 産学戦略的研究フォーラム, <http://www.iisf.or.jp/SSR/>.
- 5) Bichler, M.: The Future of e-Markets, Cambridge University Press (2001).
- 6) Box, D. et al.: Simple Object Access Protocol (SOAP) 1.1, W3C, <http://www.w3.org/TR/SOAP/> (May 2000).
- 7) Brereton, F. et al.: The Future of Software: Definition and Research Agenda, CACM, Vol.42, No.12, pp.78-84 (Dec. 1999).
- 8) Christensen, E. et al.: WSDL 1.1., <http://www.w3.org/TR/wsdl> (Mar. 2001).
- 9) Dan, A. et al.: Business-to-Business Integration with tpaML and a Business-to-Business Protocol Framework, IBM Systems J., Vol.40, No.1, pp.68-90 (2001).
- 10) 藤本隆宏ほか (編): ビジネスアーキテクチャ, 有斐閣 (2001).
- 11) Grangard, A. et al.: ebXML Technical Architecture Specification, v.1.0.4, <http://www.ebxml.org> (Feb. 2001).
- 12) Kaplan, S. et al.: E-Hubs: The New B2B Marketplaces, HBR (May-Jun., 2000), [有賀裕子 (訳), e-ハブ: B2B市場のビジネスモデル, ハーバードビジネスレビュー, 2000年12月号, pp.88-97].
- 13) Leymann, F.: Web Service Flow Language (WSFL1.0), <http://www-4.ibm.com/software/solutions/webservices/pdf/WPS.pdf> (May 2001).
- 14) Menasce, D. A. and Almeida, V. A. F.: Scaling for E-Business, Prentice Hall PTR (2000).
- 15) Oram, A. (ed.): Peer-To-Peer, O'Reilly (2001).
- 16) RosettaNet, <http://www.rosettanet.org>
- 17) Raisch, W. D.: The Marketplace, McGraw-Hill (2001).
- 18) Shim, S. S. Y. et al.: Business-to-Business E-Commerce Frameworks, IEEE Computer, Vol.33, No.10, pp.40-47 (Oct. 2000).
- 19) Thatte, B.: XLANG, [http://www.getdotnet.com/team/xml\\_wsspecs/xlang-c/default.htm](http://www.getdotnet.com/team/xml_wsspecs/xlang-c/default.htm) (2001).
- 20) UDDI, UDDI V1 Technical White Paper, <http://www.uddi.org> (Sep. 2000).
- 21) Wang, W. et al.: E-Process Design and Assurance Using Model Checking, IEEE Computer, Vol.33, No.10, pp.48-53 (Oct. 2000).
- 22) <http://www.seto.nanzan-u.ac.jp/WebSE/>

(平成13年8月20日受付)