ソフトウェアサービス技術が開く新たな 情報システム構築モデル

青 山 幹 雄 (南山大学)

Creation of Next Generation Information Systems Based on Software Services Engineering

Mikio Aoyama (Nanzan University)

Software services, such as Web services, are emerging technologies to reuse software as services over the Internet by wrapping underlying computing models with XML. Software services are rapidly evolving and are expected to change the paradigms of both development and use of information systems. This article discusses the current status and challenges of software services technologies. We classify the software services technologies into two categories: platform and development technologies. On the platform technologies, we provide a service-oriented computing framework for publishing, finding and executing services over the Internet. The framework consists of number of technologies such as SOAP, WSDL and UDDI. Brokerage of services is discussed. On the service-oriented development technologies, we discuss two development models: business-driven and user-driven models. The software services are expected to enable more dynamic collaboration across the organizational boundaries, thus, demand us to re-think the way we organize corporation and business processes. Finally, a number of open issues are pointed out.

キーワード: 情報システム, インターネット, 電子商取引, ソフトウェア工学, XML, Web サービス

Key words: Information systems, Internet, e-Business, Software engineering, XML, Web services

1. インターネット時代の情報システム像

ソフトウェアサービス技術の全体像を示し、それに基づく情報システムの新たな構築方法の現状と今後の展望を述べる。

インターネットの普及によって、われわれは、Webブラウザや電子メールを用いて世界中のコンピュータやユーザと情報を交換できるようになった。このため、企業でも販売、製造、開発などの主要業務から会計や人事などの、いわゆるバックオフィス業務にいたるすべての業務をインターネットで行うことにより、競争力を高める試みが行われている。このように、電子商取引 (e-commerce) に留まらず、企業活動全体をインターネット上で電子的に行うことをe-ビジネスと総称している。これは、政府や行政機関の電子化を行う電子政府や社会全体の電子化、いわばe-ソサエティ、へと広がっている。e-ビジネス/e-ソサエティの実現は、インターネットを基盤とする新たな情報システムにかかっているため、情報システムの構築にも変革が求められている。

図1に情報システムの進化の流れを示す。1990年代は、企業活動全体を統合して支援する、いわゆる、企業情報シ

ステム (Enterprise Information System) の構築が大きな課題であった。ERP (Enterprise Resource Planning) は、その代表例である。しかし、インターネットの非集中アーキテクチャと開放性をもたらす「どこにでもつながる」特性は、情報システムに新たな可能性をもたらした。企業間での情報システムの連携が、金融や流通などの特定業界内の個別の連携から、業界を超えてオープンに連携できるようになってきた。このため、グローバルなネットワーク上で企業間にまたがる情報システムの構築が新たな課題となっている。例えば、SCM (Supply Chain Manage-

図1 企業情報システムの進化

ment) では、顧客から流通、販売、製造を通したプロセス全体を最適化する情報システムによって、個別企業内では達成できなかった相乗効果を発揮し、いっそうの競争力の向上が期待できる。しかし、このような情報システムの構築は、対象範囲の広がりや複雑度の増大によって、いっそう困難になると考えられる。

このため、インターネット上でe-ビジネスを実現する情報システムはどうあるべきかが問われてきた⁽⁶⁾。例えば、90年代に普及した分散オブジェクト環境は、クライアント/サーバーアーキテクチャ上でオブジェクト指向に基づいてソフトウェアを実現する技術である。しかし、特定のクライアント群が特定のサーバに処理を依頼する、密結合で閉じたアーキテクチャであるため、非集中性や開放性が本質であるインターネットのアーキテクチャとは必ずしも適合しない。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。

2. ソフトウェアサービス技術とは

2.1 e-ビジネスを実現するソフトウェアサービス技術

非集中で開放的なインターネット上でe-ビジネスを支援するソフトウェアを実現するためには、企業間で異なるプログラムのインタフェースやデータ定義をより高い抽象度で整合をとって、相互運用を実現する必要がある。従来の分散オブジェクト環境でも、プログラミング言語やコンピュータの物理的位置によらずインタフェースの整合をとることは可能であった。しかし、異なる分散オブジェクト環境間の相互運用は困難であった。更に、分散オブジェクト環境のプログラム間プロトコルは、バイナリデータであるので、企業間でファイアウォールを越えて連携することは困難であった。

これに対し、XML (eXtensible Markup Language)⁽⁷⁾を基礎としてソフトウェアの相互運用のための新たな基盤技術が開発された。XMLは、それ自身で語彙を定義できる自己記述性によって、言語を定義・拡張できる。従って、基盤環境とは独立にプログラムのインタフェースやデータの意味を定義し、インターネット上で相互運用することが可能となる。更に、テキストベースのプロトコルによって、ファイアウォールを透過して相互運用が可能となる。

XMLを用いて、業務や機能単位でプログラムのインタフェースを隠ぺいし、インターネット上で再利用可能とし

図2 ソフトウェアサービス技術とは

たものを「サービス」と呼ぶ。

サービスはインターネット上で既に広く提供されている。例えば、図2に示すインターネット上での物品購入は、「物品検索」「価格交渉」などのプロセスからなる。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。しかし、Webページでは、人が情報を解釈しながら利用することを前提としている。XMLを用いて情報の意味定義を行えば、コンピュータ同士が直接連携して、「物品購入」のようなサービスを実現することが可能となる。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^{(2)~(4)}。

ソフトウェアサービスの考えに立つと、様々な業務をサービスとして連携することによりe-ビジネスを実現できると考えられる。

ソフトウェアサービス技術の中で、サービスを実現する様々な基盤技術を包含する体系として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⁽¹¹⁾。Webサービスを用いて、顧客が必要とするサービスを開発するための様々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て情報システムを実現する技術をサービス指向(Service Oriented)とも呼ぶ。

Webサービスは、主として企業顧客間電子商取引(B2C)や企業間電子商取引(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、P2P(ピア・ツー・ピア)⁽¹⁶⁾を対象とするサービスをピアサービスと呼ぶ。Webサービス技術とピアサービス技術には共通点が多い。

本稿では、Webサービスを中心に、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。

2.2 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3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

図3 オブジェクトからサービスへ

(注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Webサービスで統一する

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、一つあるいは複数のオブジェクトをまとめて独立して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 (Component Object Model)、Java言語によるJavaBeansとEJB (Enterprise JavaBeans)がある。これらのコンポーネントの再利用は、その実体を組み込むため密結合となり、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などの高価なパッケージソフトウェアの機能を遠隔共同利用できる。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⁽¹³⁾。ソフトウェアサービス技術はコンポーネント⁽¹⁾技術がインターネット上で遠隔に再利用できるように進化したものといえる。

もう一つの流れは、SCMやe-マーケットプレイスなど、インターネット上で企業や業種を超えて商取引を行う情報システムの開発が要求されていることにある。レガシーEDIと呼ばれる従来のコンピュータ間の電子データ交換 (EDI: Electronic Data Interchange) では、ユーザや業種ごとに個別に取引手順を定め、アプリケーションに組み込まれていた。そのため、業種や業界を越えた相互運用が困難であった。

このような技術の流れに立ってみると、ソフトウェアサービス技術は情報システムの開発に対する次のような新たな課題の解決を目的としていることがわかる。

- (1) プラットホ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ぺいし、ソフトウェアを実装しているプラットフォーム、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する。
- (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、結合することにより、非集中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：ビジネスプロセスの違いや、

図4 ビジネス・サービス・ソフトウェア

ファイアウォールを越えて、組織間の相互運用性の実現。

このようなサービス技術をとらえるため、図4に示す3階層参照モデルを著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スの組み合わせで実現する。一方、下位のソフトウェア実装技術を隠ぺいし、抽象的で相互運用性の高いモデルの役割も果たす。

3. 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5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二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と応用が進んでいる。一方、サービス開発技術は種々の試みはあるが、まだ、確立されているとは言い難く、今後の研究開発が待たれる。

以下、本稿ではサービス実行環境と開発技術それぞれの全体像を示す。

4. ソフトウェアサービス実行環境

4.1 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6と表1に示す。多様な技術を階層ごとに段階的に研究・開発している点に特徴がある。このほかのフレームワークは文献(19)にサーベイがある。

4.2 サービス指向アーキテクチャ

サービス実行環境は、疎結合で動的なサービスの発見、

図5 ソフトウェアサービス技術の枠組み

図6 サービス実行環境のフレームワーク

結合、実行を実現するために適したアーキテクチャとして、図7に示すブローカーアーキテクチャをとる。これをサービス指向アーキテクチャ SOA (Service-Oriented Architecture) と呼ぶ。サービスの利用者をリクエスタ、サービスの提供者をプロバイダと呼ぶ。仲介者であるブローカーがリクエスタの要求に応じて、適切なサービスを探索し、起動することによって、リクエスタとプロバイダの間は疎結合となり、サービスの動的な提供も可能になる。

図7に示すように、サービスブローカーには2種類ある。一つは、ディレクトリを探索し、必要なサービスを探索し、リクエスタに通知する。もう一つは、高度な仲介機能を持つブローカーである。例えば、サービスの組み合わせや

表 1 サービス実行環境の主要素技術

技術	内容	開発	標準化
SOAP/XMLP	XML ベースの軽量手続き呼び出しプロトコル	Microsoft 他	W3C:V1.1(2000 年 5月),V1.2 WD(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	UDDI Initiative (Ariba, IBM, Microsoft 他) V1(2000 年 9 月), V2(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
Rosetta Net	SCM のビジネスプロセス定義と実行	RosettaNet (Intel 他電子部品業界)	—

図 7 サービス指向アーキテクチャ

図 8 サービス指向アーキテクチャと既存アーキテクチャ

ルーティング、QoS (Quality of Service) の保証や品質保証、セキュリティなどの要求に対応する。サービスブローカーはサービス提供の価値を高める鍵となる技術である。

図8にサービス指向アーキテクチャの実行を従来のWeb ページ、分散オブジェクト環境の実行と合わせて示す。リクエスタはWeb ブラウザの場合とほかのサーバの場合がある。分散オブジェクト環境と比べ、プラットフォームによらず、ファイヤウォールを越えた相互運用が実現できる点で企業間連携に適用できるなど、適用範囲が広がる。また、コンピュータ同士が直接相互運用できる点で、従来の人手による Web ページの利用と異なる。

4.3 サービスの記述と探索・実行

(1) サービスの記述

サービスのインタフェース記述言語として WSDL (Web Services Description Language) が W3C (World Wide Web Consortium) で標準化されている⁽⁹⁾。サービスを実現するプログラミング言語や実装方法を隠ぺいする。図9に WSDL の構造を示す。右端のプログラムを隠ぺいしてサービスとして提供するためのラッパーとなる。左端のリクエスタからみると、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。バインディングはリクエスタからの呼び出しプロトコルを任意に組み合わせることができる仕組みである。HTTP だけでなく、MIME によって電子メールの添付ファイルとしてサービスの要求を記述すると、非同期な要求も可能となる。

SOAP (Simple Object Access Protocol) は XML を用いたプラットフォーム独立な遠隔手続き呼び出し (Remote Procedure Call) プロトコルである^{(6)(注2)}。ヘッダ (header) と本体 (body) からなる。本体に XML でプログラムの呼び出しなどを記述して送受信する。

インタフェースが相互運用可能であるためには、リクエスタとプロバイダが同じ語彙を共有する必要がある。そのため、語彙とそのスコープを制御できる必要がある。これは、XML Schema の XML 名前空間 xmlns (XML name space) を用いる。

図 9 WSDL の構造

(注2) SOAP の高度化として XMLP (XML Protocol) が検討されている

(2) サービスの公開・探索

UDDI (Universal Description, Discovery and Integration) はサービスを公開し、検索するためのディレクトリである⁽²¹⁾。インターネット上のビジネスやサービスの電話帳になる。その概念データモデルも、表2に示すように、電話帳に似せてある。ここで、グリーンページが新しい概念で、直接コンピュータ間でサービスを連携するための情報を提供する。

図10にUDDI内のデータ構造を示す。ビジネスやサービスを検索する鍵となる分類情報を登録できる。業種分類は米国のNAICS (North American Industry Classification System)、製品分類はUNSPSC (UNiversal Standard Products and Services Classification)、地理的位置はISO 3166を用いている。

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⁽²¹⁾。

今後、企業内や業界内など様々な形態のUDDIの運用も考えられている。

(3) サービスの組み合わせ

Webサービスを組み合わせ、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により実行時に動的にサービスを探索し、組み合わせることが可能となる。このようなサービス

表2 UDDI V1の概念データモデル

分類	目的	内容
ホワイトページ	ビジネスの公開, 検索	ビジネス情報(businessEntity): サービスプロバイダ(企業, 組織)毎の属性(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	サービスの公開, 検索	サービス情報(businessService): サービスの毎の属性とバインド情報への参照など。サービスは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	サービスの結合	バインド情報(bindingTemplate): サービス毎に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。

図10 UDDIのデータ構造

の組み合わせを記述する言語として、WSFL (Web Services Flow Language)⁽¹⁴⁾や、XLANG⁽²⁰⁾が提案されている。

4.4 ebXMLとRosettaNet

ebXML (e-business XML)⁽¹²⁾やRosettaNet⁽¹⁷⁾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。更に、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⁽¹⁰⁾。

また、RosettaNetは、パソコンなどの電子部品業界が主導して、SCMの構築など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。

5. ソフトウェアサービスによる情報システム開発技術

5.1 サービス指向情報システム開発技術の枠組み

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。例えば、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かった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。

5.2 サービス指向情報システム開発方法論

本稿では、サービス開発技術を図11に示す二つの視点に分けて紹介する。

(1) 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一つはSCMなどの組織間で連携する情報システムの構築にある。従来、ビジネスモデルごとに個々に開発してきた情報システムを、サービスを組み合わせ、効率的に実現することが期待できる。

例えば、図12はオークションなどの異なるビジネスモデル⁽⁵⁾⁽¹⁸⁾の構造を抽出したものである。これらは、構造

図11 サービス開発技術の枠組み

図 12 ビジネスアーキテクチャパターン

図 13 ユースケースによるユーザのモデル化

的にはブローカーアーキテクチャである。従って、サービス指向アーキテクチャと親和性が高いので、サービスを組み合わせることで実現できると考えられる。

(2) ユーザモデル駆動型

ユーザモデル駆動型は B2C といった電子商取引などにおいてユーザの振る舞いを中心に設計する方法である。これには、次の二つのアプローチがある。

- (i) ユーザとサービスの相互作用モデル：例えば、ユーザがサービスを利用する手順を図 13 に示す UML のユースケースとそのシナリオで記述する
- (ii) 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、ユーザの振る舞いをグラフ上の状態遷移としてモデル化する⁽¹⁵⁾

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

5.3 システム検証技術

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、セキュリティの確保はいつそう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、モデル化や設計の段階で構造的な特性を検証する試みが行われている⁽²²⁾。

6. 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、次の二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

表 3 Web サービス技術の製品化

ベンダ	Web サービス基盤技術
IBM	Web サービス
Microsoft	.NET フレームワーク
Oracle	Oracle Dynamic Services
Sun Microsystems	ONE (Open Net Environment)

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。例えば、Microsoft の .NET (ドットネット) My Service 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 8 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。

7. 実務への応用

Web サービス実行環境の製品化例を表 3 に示す。このほか Web サービスのポータルサイトの提供 (例えば、www.xmethods.net) など、Web サービスの特性を活かした新たなビジネスも試みられている。

8. 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。技術面でもトランザクション処理など、基幹業務へ適用するうえで課題がある。実行環境・開発技術と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、上位のビジネスの構造やモデル化の研究も必要である⁽¹¹⁾。

一方、ソフトウェアサービス技術によって、情報システムのあり方、更にそれを利用する組織や業務のあり方が問われる⁽¹³⁾。

例えば、企業ではコアコンピタンスとならない業務はサービスとしてアウトソースし、組織と情報システムの両面から再構造化が起こる可能性がある。

このような背景から、著者らは、ソフトウェアサービス技術の調査・研究⁽⁴⁾、ものづくり・IT 融合化基盤環境への応用⁽²³⁾、国際ワークショップの開催⁽²⁴⁾などを行っている。

9. ま と め

ソフトウェアサービス技術は XML をベースとして、インターネット上でソフトウェアコンポーネントを再利用する技術である。分散オブジェクト環境が実現できなかった異種プラットフォーム間の非集中でオープンな相互運用を可能とし、インターネットのもたらす本質的な特性を活用し、課題の解決を目指すものである。今後の情報システム構築の基盤技術となる。また、サービスディレクトリなど、技術とビジネスの両面で重要な役割をはたすと期待される新しい技術も開発されている。

今後、我が国の技術者、研究者が本分野の研究・開発へ参画されることを期待する。

最後に、本稿をまとめる機会をいただいた、(株)日立製

作所の船橋誠壽氏に感謝する。

(平成 14 年 3 月 8 日受付)

文 献

- (1) 青山幹雄・中武司・向山 博 (編著) : コンポーネントウェア, 共立出版 (1998)
- (2) M. Aoyama, Intelligent Software Services over the Internet, H. Kangassalo, et al. (eds.) : "Information Modeling and Knowledge Bases", *IX, IOS Press*, pp. 128~135 (2000)
- (3) 青山幹雄 : 「ソフトウェアサービス技術へのいざない」, 情報処理, Vol. 42, No. 9, pp. 857~862 (2001)
- (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, 戦略的研究フォーラム, <http://www.iisf.or.jp/SSR/>
- (5) M. Bichler : The Future of e-Markets, Cambridge University Press (2001)
- (6) D. Box, et al. : Simple Object Access Protocol (SOAP) 1.1, W3C, May (2000) <http://www.w3.org/TR/SOAP/>
- (7) T. Bray, et al. (eds.) : Extensible Markup Language (XML) 1.0, 2nd ed., W3C (2000) <http://www.w3.org/XML/>
- (8) F. Brereton, et al. : "The Future of Software: Definition and Research Agenda", *CACM*, Vol. 42, No. 12, pp. 78~84 (1999)
- (9) E. Christensen, et al. : WSDL 1.1, W3C, (2001) <http://www.w3.org/TR/wsdl>
- (10) A. Dan, 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)
- (11) 藤本隆宏・ほか(編) : ビジネスアーキテクチャ, 有斐閣, (2001)
- (12) A. Grangard, et al. : ebXML Technical Architecture Specification, v. 1.0.4 (2001) <http://www.ebxml.org>
- (13) J. Hagel III and J. S. Brown, Your Next IT Strategy, Harvard Business Review, Oct, pp. 105~113 (2001) [西 尚久(訳), ウェブ・サービス・アーキテクチャの可能性 : 「所有から利用」への IT マネジメント, DIAMOND ハーバード ビジネス・レビュー, Dec., pp. 115~127 (2001)]
- (14) F. Leymann, Web Services Flow Language (WSFL 1.0), May (2001) <http://www-4.ibm.com/software/solutions/webservices/pdf/WPS.pdf>
- (15) D. A. Menasce and V. A. F. Almeida : Scaling for E-Business, Prentice Hall PTR (2000)
- (16) A. Oram (ed.) : Peer-To-Peer, O'Reilly (2001)
- (17) RosettaNet : <http://www.rosettanet.org>
- (18) W. D. Raisch : The Marketplace, McGraw-Hill (2001)
- (19) S. S.Y. Shim, et al. : "Business-to-Business E-Commerce Frameworks", *IEEE Computer*, Vol. 33, No. 10, pp. 40~47 (2000)
- (20) B. Thatte : XLANG, (2001) <http://www.gotdotnet.com/team/xmlwsspecs/xlang-c/default.htm>
- (21) UDDI : UDDI V 1 Technical White Paper (2000) <http://www.uddi.org>
- (22) W. Wang, et al. : "E-Process Design and Assurance Using Model Checking", *IEEE Computer*, Vol. 33, No. 10, pp. 48~53 (2000)
- (23) <http://unit.aist.go.jp/digital-mfg/>
- (24) <http://www.seto.nanzan-u.ac.jp/WebSE/>

青 山 幹 雄 (非会員) 1980 年岡山大学大学院工学研究科修士課程了。同年より 95 年まで富士通(株)勤務。86~88 年米国イリノイ大学客員研究員。95~01 年新潟工科大学情報電子工学科教授。01 年 4 月より現職。ソフトウェア開発方法論, コンポーネントウェア, ソフトウェアプロセスなどに興味を持つ。93 年情報処理学会研究賞受賞。情報処理学会, 電子情報通信学会, 日本ソフトウェア科学会, IEEE, ACM 各会員。