企業情報システムにおける連携技術

正 員 田中 哲雄*

正 員 湯本 真樹**

正 員 齋 礼***

Enterprise Information System Integration Technology

Tetsuo Tanaka*, Member, Masaki Yumoto**, Member, Rei Itsuki***, Member

In the current rapidly changing business environment, companies need to be efficient and agile to survive and thrive. That is why flexible systems integration is urgent and crucial concern for any enterprise. For the meanwhile, systems integration technology is getting more complicated, and middleware types are beginning blur for decades. We sort system integration into four different types, “Delayed Federation”, “Real-time Federation”, “Delayed Integration”, and “Real-time Integration”. We also outline appropriate technology and architecture for each type.

キーワード：企業情報システム、システム連携、インテグレーション・ブローカ、アプリケーション・サーバ、ETL ツール、BPM、エンタープライズ・ポータル、EII

Keywords：Enterprise Information System, System Integration, Integration Broker, Application Server, ETL Tool, BPM, Enterprise Portal, EII

1. はじめに

異業種からの市場参入、顧客の嗜好の変化、競合する企業同士の合併など、企業の経営環境の変化は著しく加速している。企業は、絶えず変化する経営環境に合わせて、提供する商品やサービス・業務のプロセス・組織構造・パートナとの関係を継続的に改善していかなければならない。一方、今日の企業活動は情報システムに大きく依存している。したがって、企業がその活動を継続的に改善していくために、変化に柔軟に対応できる情報システムが望まれている。

企業内には、在庫管理システム、営業支援システム、会計システム、人事管理システムなど、これまで構築されてきたさまざまな情報システムがある。各情報システムは、それぞれが開発された時代の経営環境における要求に最適にできるように、その時代の最良の技術を使って、多大な

投資により開発されており、依然として大きな価値を持っている。

このような時代の必然として、価値あるシステム資産を活用しつつ、それらシステム資産を環境の変化に柔軟に対応させる「システム連携技術」が注目されている。経営環境がダイナミックに変化すればするほど、システム連携技術の役割は大きなものになる。

とはいえ、異なる時代に異なる目的で開発されたシステムを連携させるのは容易ではない。情報システム毎に、前提とするハードウェアやOS、ミドルウェアなどのプラットフォームが異なる、あるいは、扱うデータ形式が異なるといった問題がある。システムによっては、別システムとデータをやり取りするインターフェースがないこともある。これらの問題を解決するために様々な技術が提案され、ミドルウェア製品がリリースされている。連携の目的や運用形態によって、利用すべき技術やミドルウェアは異なるが、技術やミドルウェアの機能の多様化により、適切な技術・ミドルウェアの選択は困難になりつつある。

本稿では、システム連携をいくつかのタイプに分類し、そのタイプ毎に採用すべきアーキテクチャとそれを支える要素技術を解説する。

2. システム連携のタイプ

システム連携の実現方法を整理するために、本章ではシ

* (株)日立製作所 システム開発研究所
〒215-0013 神奈川県川崎市麻生区王禅寺 1099
Systems Development Laboratory, Hitachi, Ltd.

** 1099 Ohzenji, Asao-ku Kawasaki-shi, Kanagawa 215-0013
近畿大学 理工学部 電気電子工学科
〒577-8502 大阪府東大阪市小若江 3-4-1
Kinki University

*** 3-4-1 Kowakae Higashi-Osaka, Osaka 577-8502
広島国際大学 社会環境科学部 情報通信学科
〒737-0112 広島県呉市広古新開 5-1-1
Hiroshima International University
5-1-1 Hirokoshingai, Kure-city, Hiroshima 737-0112

図1 フェデレーション型 vs. インテグレーション型

Fig. 1. Federation vs. integration.

図2 ディレイド型 vs. リアルタイム型

Fig. 2. Delayed vs. real-time.

表1 システム連携のタイプ

Table 1. Types of system integration.

運用形態による分類 目的による分類	ディレイド型	リアルタイム型
	<ul style="list-style-type: none"> データ転送に重きをおいた運用 ストア&フォワード、パブリッシュ&サブスクライブ 変更を遅れて反映 大量かつ複雑なデータの受け渡し 	<ul style="list-style-type: none"> ユーザ・インタラクションに重きをおいた運用 リクエスト/リプライ 変更を即座に反映 少量かつ単純なデータの受け渡し
フェデレーション型	[ディレイド・フェデレーション型] <ul style="list-style-type: none"> データ・レプリケーション あるシステムのデータを、形式を変換しながら他のシステムに移行する データ・シンクロナイゼーション(非リアルタイム) 複数のシステムに存在する同じ事象を表すデータを同期化する 	[リアルタイム・フェデレーション型] <ul style="list-style-type: none"> データ・シンクロナイゼーション(リアルタイム) 複数のシステムに存在する同じ事象を表すデータを同期化する
インテグレーション型	[ディレイド・インテグレーション型] <ul style="list-style-type: none"> アプリケーション・コーディネーション 複数のステップからなる単一の業務をサポートするために、ステップに対応する業務システムを統合 BtoBインタラクション 複数企業間にまたがるビジネスプロセスの自動化 	[リアルタイム・インテグレーション型] <ul style="list-style-type: none"> エンタープライズ・ポータル 複数のWebページを一面にレンダリング アプリケーション・コンポジション 複数のコンポーネントを統合して単一のAPを構成 データ・アグリゲーション 複数のデータを集約して統合データ・ビューを構成

システム連携を、その目的および運用形態の二つの軸でタイプ分けし、それぞれの特性を述べる。

システム連携には、大きく分けて2つの目的がある。一つは、独立の業務のために独立に開発された情報システム群に重複して存在するデータの整合性確保である。これを「フェデレーション型」と呼ぶ。異なる目的で異なる時期に開発されたシステムには、同じものごとを表すデータが重複して格納されていることが多い。それらの整合性を自動的に確保できれば、データの再入力の手間や入力ミスなどによる不整合が軽減される。もう一つの目的は、新たなアプリケーションによる新たな価値の提供である。これを「インテグレーション型」と呼ぶ。既存の情報システムを複数組み合わせる新たなサービスを提供する。(図1参照)

一方、システム連携の運用形態という観点から眺めると、ユーザ・インタラクションに重きをおき、ユーザからの要求に対し、他のシステムとリアルタイムに連携しながら即座に処理して応答を返す「リアルタイム型」と、データの転送に重きをおき、ユーザからの要求やビジネスイベントに対する処理が即時ではなく、時間をおいてなされる「ディレイド型」がある。(図2参照)

以上を纏めると、システム連携のタイプは、「ディレー

ド・フェデレーション型」、「リアルタイム・フェデレーション型」、「ディレイド・インテグレーション型」、「リアルタイム・インテグレーション型」の4つに分類できる。それぞれのタイプの特徴と例を表1に纏める。

ディレイド・フェデレーション型の例としては、ERP (Enterprise Resource Planning) パッケージの販売実績、購買実績などのオペレーションデータを、BI (Business Intelligent) ツールの入力とするなどの「データ・レプリケーション」や、ERP パッケージと CRM (Customer Relationship Management) パッケージの顧客データを同期化するという「データ・シンクロナイゼーション」がある。

ディレイド・インテグレーション型の例としては、インターネット上で提供されるホテルの宿泊予約サービスや飛行機のフライト予約サービスなど各種サービスを統合しワンストップで提供するという「アプリケーション・コーディネーション」や、複数企業間にまたがりビジネス・プロセスを自動化する「BtoB インタラクション」がある。

リアルタイム・フェデレーション型の例は、ディレイド・フェデレーション型の例と同様に「データ・シンクロナイゼーション」であるが、例えば、株式市場システムなどの

ように、データの鮮度やトランザクション管理に厳しい要求があるときリアルタイム・フェデレーション型となる。

リアルタイム・インテグレーション型の例としては、企業内の複数の Web アプリケーションを一つの画面にまとめた「エンタープライズ・ポータル」や、販売管理システム、生産管理システム、購買システムなど関連する複数のシステムを統合して、納期回答システムを構築するといった、「アプリケーション・コンポジション」、または、金融機関において、預金、投資信託、住宅ローンなどの金融商品の単位で管理されている顧客情報を、顧客というキーで統合して仮想的なテーブルを生成するといった「データ・アグリゲーション」などがある。

ここで注意したいのは、ある業務向けにサービスを構築する場合、それはリアルタイム型でもディレード型でも実現できるということである。前述の例でも、リアルタイム型の例として挙げたものをディレード型として実現することもできるし、その逆も可能である。例えば、旅行代理店サービスの場合、エンドユーザからの要求があった時点で、リアルタイムに宿泊予約サービスやフライト予約サービスに要求を出し、その応答を即座にマージして旅程表としてユーザに返すリアルタイム型として実現することもできるし、ユーザからの要求受付時には、受け付けた旨のみを一旦ユーザに返し、宿泊予約、フライト予約の処理が終了し旅程表が完成した時点で、メールなどでユーザに通知し、次にユーザが旅行代理店サイトにアクセスした際に結果である旅程表を提示するディレード型として実現することもできる。

リアルタイム型/ディレード型の選択は、ユーザ・インタラクションの重要度、要求されるデータの鮮度、トランザクションの単位、やり取りされるデータの量、および、コストなどを総合的に勘案してなされる。

3. システム連携の実現技術

フェデレーション型では、あるシステムから別のシステムにデータを転送する仕掛けが必要になる。また、インテグレーション型を実現するには、フェデレーション型の実現技術に加え、データやアプリケーション・ロジック、プレゼンテーションなどの統合の仕掛けが必要になる。以下、フェデレーション型のための仕掛けをフェデレーション・レイヤ、インテグレーション型のために特に必要となる仕掛けをインテグレーション・レイヤと呼ぶ。図 3 に示すように、インテグレーション・レイヤはフェデレーション・レイヤの機能を前提とする。

それぞれのレイヤ向けにベンダー各社からミドルウェア製品が提供されているが、ミドルウェアを用意するだけではシステム連携は実現できない。フェデレーション・レイヤでのデータの転送には、転送されるデータの内容を送る側と送られる側で共通化しておく必要があるし、統合レイヤでの、BtoB インタラクションでは、複数の組織が別々にビジネス・プロセスを管理するので、そのビジネス・プロ

図3 連携技術の技術スタック

Fig. 3. Technology stack of system integration.

セスを共通化しておく必要がある。これらデータの内容やビジネス・プロセスをそれぞれのレイヤのコンテンツと呼ぶことにする。

以下、両レイヤのミドルウェアとコンテンツを紹介する。

〈3・1〉 フェデレーション・レイヤのミドルウェア

フェデレーション・レイヤのミドルウェアには、MOM(Message Oriented Middleware)、ORB(Object Request Broker)、TP (Transaction Processing) モニタなど、多くの製品がベンダーから提供されている⁽¹⁾⁽²⁾。MOMとは、メッセージの確実な配信機能を提供するミドルウェアである。メッセージをMOMに送信すると、MOMは、それを一旦メッセージのキューに置き、その後、宛先に送信する。ORBは、異なるマシン上に分散して存在するオブジェクト間で、処理の要求やその応答などのメッセージをやりとりする際に用いられるミドルウェアである。TPモニタは、複数の処理をまとめてトランザクションという単位で監視し、すべて成功かすべて失敗のいずれかであることを保証するミドルウェアである。

現在主流となっているフェデレーション・レイヤのミドルウェアは、MOMの機能をベースとしたインテグレーション・ブローカ⁽³⁾⁽⁵⁾とETL(Extract Transform Load)ツール⁽⁶⁾⁽⁷⁾、および、ORBやTPモニタの機能を包含するアプリケーション・サーバ⁽³⁾⁽⁴⁾である。以下、それぞれについて説明する。

(1) インテグレーション・ブローカ

インテグレーション・ブローカは、MOMの機能に加え、データ変換やインテリジェント・ルーティングなどの機能をもつ。

連携対象となるアプリケーションとは、データ形式やプロトコルを変換するアダプタによって接続する。パッケージ・アプリケーションやBtoBの各種標準プロトコル用のアダプタが豊富に提供されるのが特徴である。

インテリジェント・ルーティングとは、アプリケーション間で送受信されるメッセージの内容によって、メッセージの送信先を決定する仕掛けである。これにより、例えば、受注処理において、通常の受注オーダは、受注管理システムのみで処理し、会員番号と注文価格がある条件を満たす場合は、顧客管理システムにもメッセージを渡すといった

処理が容易になる。

(2) ETL ツール

ETL ツールは、基幹系業務システムなどのデータ・ソースからデータを抽出 (extract) し、必要に応じてデータを変換 (transform) し、データ・ウェアハウスなどのターゲット・システムにそのデータをロード (load) するミドルウェアである。一般に、不正なデータを排除したり標準の形式にデータを修正したりするデータ・クレンジング機能を持ち、基幹業務システムで発生したデータを整形してデータ・ウェアハウスへ移す際に利用される。

(3) アプリケーション・サーバ

アプリケーション・サーバは、ユーザからの要求を受け付けて、データベースなどの業務システムの処理に橋渡しする機能を持つ。クライアントとして Web ブラウザが用いられ、HTTP(Hypertext Transfer Protocol)で通信を行なう。一般に、Web サーバとしての機能、EJB™(Enterprise JavaBeans™)や Servlet, JSP(JavaServer Pages™)などのプログラムの実行環境、データベースや他の分散オブジェクトへの接続機能、複数の処理を連結するトランザクション管理機能などをもつ。更に、大規模なユーザに対応するため、ロード・バランシングや、データベースの接続プール、フェールオーバー等の機能を備えているものが多い。

〈3・2〉 統合レイヤ 複数のシステムの統合には、ビジネス・プロセスの統合、データの統合、プレゼンテーションの統合がある。それぞれ、BPM(Business Process Management)⁽⁵⁾⁽⁸⁾⁽⁹⁾, EII(Enterprise Information Integration)⁽⁷⁾, エンタープライズ・ポータル⁽¹⁰⁾⁽¹¹⁾として注目されている。

(1) BPM

BPM とは、予め定義されたビジネス・プロセスに従って、複数の業務システムを統合・制御し、また、それをモニタリングする仕組みであり、ワークフロー・システムの発展系といえる。従来のワークフロー・システムが、人の作業や承認などのビジネス・プロセスを定義・管理するのに対して、BPM では、業務システムとの連携を重視する。

図 4 にあるように、ビジネス・プロセスは、「商品の受注」や「顧客からの問い合わせ」などのビジネスイベントによって開始され、ビジネス・プロセスの定義に従って、業務システムや人に処理を実行させる。また、ビジネス・プロセスをモニタリングすることによって、ボトルネックを分析し、ビジネス・プロセスの継続的な改善につなげられる。業務の流れに変更が生じて、ビジネス・プロセスの定義を変更するだけでなく、個々の業務システムを改変する必要がないのが特徴である。

ビジネス・プロセスの定義には、GUI の定義画面を用いる。ビジネス・プロセス中の各アクティビティを矩形などで表示し、各矩形を業務システムに対応付ける。その矩形を線で結ぶことで、業務の流れを定義する。

(2) EII

EII とは、データベースやスプレッドシート・ファイルな

図 4 BPM
Fig. 4. BPM.

図 5 EII
Fig. 5. EII.

どにリアルタイムにアクセスし、アプリケーションのニーズにマッチした仮想的なビューを構成する技術である。図 5 に示すように、データ・ソースのメタデータを管理し、それを利用して、統合ビューや統合キャッシュの機能を実現する。

ユーザは、企業内に散在するデータ・ソースに統一的かつリアルタイムにアクセスできる。このとき、データの参照だけではなく、更新や削除をトランザクショナルに行なうことができる。また、そのためにデータ・ソースを変更する必要もない。

例えば、本社と支社で独自に構築してきたデータベースの統合や、企業が合併したときの全社データベースの構築などに有効な手段になる。

(3) エンタープライズ・ポータル

エンタープライズ・ポータルは、基幹業務システムやグループウェア、各種データベース、更には、社外の情報サイトなど、企業内外に存在する情報資源・サービスを集約して、一つの画面から効率的にアクセスするための Web サイトである。例えば、経営幹部向けのポータルでは、経理システムや営業支援システムなどから経営者にとって重要

図6 エンタープライズ・ポータル
Fig. 6. Enterprise portal.

なデータを抽出して表示する。

エンタープライズ・ポータルは、図6に示すように、情報資源やサービスに対応したポートレットと呼ぶHTMLベースのコンポーネントを組み合わせて構築する。ポータル・サーバは、複数のポートレットからポータル画面を構成するレンダリング、ユーザごとに適した情報を提示するパーソナライズ、ID/パスワードの入力の手間を省くシングル・サインオン、携帯電話やPDAなど多様な端末に対応するトランスコーディングなどの機能を備える。

〈3・3〉コンテンツ システム連携には、連携するシステム同士が共通に解釈できるコンテンツが必要となる。一企業内のシステム連携では、その企業の中で独自に共通のコンテンツを定めればよいが、企業間の連携では、コンテンツの標準化が必要となる。企業間の連携を行う場合、業界ごとにビジネス・プロセスが大きく異なり、また、やり取りするデータの内容も異なるので、コンテンツの標準化は業界ごとに進められている。

(1) フェデレーション・レイヤのコンテンツ

フェデレーション・レイヤでは、企業間でやり取りされるデータ（メッセージ）の構文と語彙（辞書）を規定する必要がある。

例えば、電子機器業界のBtoB標準であるRosettaNetTM(13)(14)では、構文として、XML (eXtensible Markup Language) をベースとしたPIP® (Partner Interface Process®) メッセージを規定しており、語彙としては、企業コードにD-U-N-S® ナンバー (The Data Universal Numbering System)、製品コードにGTIN (Global Trade Item Number) という業界標準を採用している。

RosettaNetの他に、化学業界のCIDXTM (Chemical Industry Data eXchange)、流通業界のJEDICOS® (Japan EDI for Commerce Systems)-XMLメッセージなど、業界毎に多数の標準が策定されている(15)(16)。

(2) 統合レイヤのコンテンツ

BtoBインタラクションのように複数の企業が別々に統合レイヤのミドルウェアを運用する場合、企業間でどのよ

うなビジネス・プロセスでインタラクションするかを予め取り決めておく必要がある。

例えば、前述のRosettaNetでは、PIPと呼ぶビジネス・プロセス定義を、見積もりや発注などの業務単位毎に詳細に規定している。

更に、ビジネス・プロセスを記述する言語も標準化が進められており、OASIS (Organization for the Advancement of Structured Information Standards) が推進する、BPEL4WS (Business Process Execution Language for Web Services) (19)、ebXML BPSS (Business Process Specification Schema) (17)、W3C® (World Wide Web Consortium) が推進するWS-CDI (Web Service Choreography Interface) (18) などがある。

4. システム連携のタイプと実現技術の関係

システム連携に必要となる実現技術は連携のタイプごとに異なる(3)(12)。各タイプの実現技術を図7にまとめる。

ディレド型はデータ転送に重きをおいておりユーザとのインタラクションはないので、バッチまたは即時のファイル転送/メッセージ転送が実現の主役となる。また、大量かつ複雑なデータの受け渡しをすることが多く、辞書や変換ルールに基づく高度なデータ変換機能が必要となる。3.で紹介したミドルウェアとしては、MOMの機能、および、データ変換機能を持つインテグレーション・ブローカとETLツールが利用される。

リアルタイム型はユーザ・インタラクションに重きをおいているので、即時のリクエスト/リプライが実現の主役となる。更に書き込みのロールバックが必須の場合は、トランザクション処理の機能が必要となる。3.で紹介したミドルウェアの中では、主にORB/TP monitorの機能をもつアプリケーション・サーバが利用される。

フェデレーション型には、インテグレーション・ブローカ、ETLツール、アプリケーション・サーバなどのフェデレーション・レイヤのミドルウェアを用いる。表1に挙げた例では、データ・レプリケーションはETLツールを、非リアルタイムのデータ・シンクロナイゼーションはインテグレーション・ブローカを、リアルタイムのデータ・シンクロナイゼーションはEIIツールを用いる。

インテグレーション型では、フェデレーション・レイヤのミドルウェアに加え、統合の種類によって更に、インテグレーション・レイヤをサポートするミドルウェアが必要となる。アプリケーション・コーディネーションとアプリケーション・コンポジションでは、高度なフローコントロールが必要となる。これらは、それぞれ、BPMツールのビジネス・プロセス管理機能、アプリケーション・サーバ上で稼動する統合ロジック内に埋め込まれたマイクロ・フローとして実現される。同様に、エンタープライズ・ポータルとデータ・アグリゲーションは、それぞれ、ポータル・サーバにおけるレンダリングエンジン、EIIツールにおける仮想統合ビューによって実現される。

図7 システム連携のタイプと連携技術の関係

Fig. 7. Relationship between integration types and technology.

5. システム連携の最近の動向

これまで、システム連携を4つのタイプに分類し、それぞれに適する実現技術を述べてきたが、ベンダー各社が提供する製品をみると、そのカテゴリの境界は曖昧になってきている。例えば、インテグレーション・ブローカはもともとサーバ間のディレード型の連携機能を提供してきたが、今日では、リクエスト/リプライ型通信やWebアプリケーション作成などのアプリケーション・サーバの機能を提供するようになってきている。また、インテグレーション・ブローカ、アプリケーション・サーバ、ポータル・サーバ、BPM ツールなど、システム連携に必要な機能を集めてスイート製品として提供するベンダーが増えている。

このように製品の機能が拡充され高価になっていく一方で、より安価で軽量のESB(Enterprise Service Bus)⁽²⁰⁾⁽²¹⁾という製品カテゴリが注目されている。ESBは、XML、Webサービス、JMS(Java Message Service)、JCA(J2EE™ Connector Architecture)などの業界標準を採用し、限定的ではあるがデータ変換、インテリジェント・ルーティング、MOMなどの機能を提供する。BPMやポータル、高度な変換やアダプタ開発ツールはもたない。

また、システム連携の基盤としてWebサービスの果たす

図8 Webサービスの技術スタック

Fig. 8. Technology stack of web services.

役割が大きくなるのは間違いない。図8に示すような各種標準化がOASISなどの標準化団体にて、急ピッチで進められており⁽²²⁾、3.で説明した各ミドルウェアがそれらの標準に対応しつつある。フェデレーション・レイヤのミドルウェアはSOAPでのデータの送受信をサポートし、統合レイヤのポータル・サーバがWSRP(Web Services for Remote Portlets)をサポートしている。BPMについても同様に、BPEL4WSの標準化が完了し次第、ミドルウェアがそれに対応していくと考えられる。

Web サービスについては、セキュリティや信頼性、トランザクションの扱いなど QoS (Quality of Service) に関する不安が語られることが多いが、この部分についても、WS-Security, WS-Reliability, WS-CAF(Web Services Composite Application Framework)など、対応する仕様の標準化活動が活発に行われている⁽¹⁹⁾⁽²²⁾。また QoS と並んで Web サービスの普及に不可欠とされてきた管理についても、OASIS にて WSDM(Web Services Distributed Management)として標準化が進められている。

6. おわりに

この解説では、システム連携を目的と運用という視点から、フェデレーション型/インテグレーション型とリアルタイム型/ディレード型という二つの軸で四つのタイプに分類した。システム連携に利用される技術を解説し、必要となる技術やアーキテクチャはタイプ毎に異なることを述べた。

ミドルウェア製品のカテゴリには、インテグレーション・ブローカ、アプリケーション・サーバ、ETL ツール、BPM ツール、EII ツール、ポータル・サーバなどがあるが、それぞれのカテゴリの製品が、別のカテゴリの機能をサポートしており、その境界は曖昧になってきている。この解説が、システム連携に取り組む際、その連携の本質を見極め、適用すべき技術の検討の一助になれば幸いである。

謝 辞

この解説は、電気学会「企業情報システムにおける連携技術調査専門委員会」の活動の一環として執筆したものである。熱心に議論いただいた薦田委員長、ならびに、委員の方々に深謝いたします。

(平成 16 年 2 月 20 日受付)

文 献

- (1) P. A. Bernstein : "Middleware: A Model for Distributed System Services", Commun. of the ACM, Vol.39, No.2, pp.86-98 (1996)
- (2) W. Emmerich : "Software engineering and middleware: a roadmap", Proc. conf. The future of Software engineering, pp.117-129 (2000)
- (3) J. Thompson : "Middleware: Foundation for the Enterprise Nervous System", Gartner Web Services & Appl. Integration Conf. 2003 (2003)
- (4) 「オープンセミナー A/Pサーバ導入時のチェック・ポイント」, 日経オープンシステム, 2001.9 pp.220-227 (2001)
- (5) E. Roch : "Application Integration: Business and Technology Trends", eAI Journal, August 2002, pp.34-39 (2002)
- (6) G. Pancha : "EAI and ETL: Meeting the Integration Challenge", eAI Journal, July 2002, pp.26-28 (2002)
- (7) W. Kernochan : "Enterprise Information Integration - The New Way to Leverage E-information - (Second Edition)", Aberdeen Group (2003)
- (8) P. Fremantle, S. Weerawarana, R. Khalaf : "Developing and integrating enterprise components and services: Enterprise services", Commun. of the ACM, Vol.45, No.10 (2002)
- (9) 「注目を集める BPM ツール」, 日経コンピュータ, 2001.7.2, pp.38-40 (2001)
- (10) T. K. Hazra : "Building enterprise portals: principles to practice", Proc. of the 24th international conf. on Software engineering, pp.623-633 (2002)
- (11) 小池 博・田中哲雄 : 「e ビジネスにおける XML -XML/EDI とエンタープライズ・ポータル」, 情報処理, 42, 7, pp.648-653 (2001-7)
- (12) M. Stonebraker : "Too much middleware", ACM SIGMOD Record, Vol.31, No.1, pp.97-106 (2002)
- (13) 大内博志 : 「RosettaNet」, 情報処理, 42, 9, pp.890-894 (-9)
- (14) RosettaNet Standards, <http://www.rosettanet.org/standards>
- (15) P. Kotinurmi, J. M. Nurmilaakso, H. Laesvuori : "Standardization of XML-based e-business frameworks", Standard Making: A Critical Research Frontier for Information Systems MISQ Special Issue Workshop, pp.135-146 (2003)
- (16) JEITA : 「XML フォーマットのイエローページ」, <http://it.jeita.or.jp/eltech/XML/>
- (17) "Business Process Specification Schema Version1.01", UN/CEFACT and OASIS, 2001
<http://www.ebxml.org/specs/ebBPSS.pdf>
- (18) Web Service Choreography Interface (WSCI) 1.0, W3C Note (2002) <http://www.w3.org/TR/wsci>
- (19) F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana : "Service-oriented computing: The next step in Web services", Commun. of the ACM, Vol.46, No.10 (2003)
- (20) R. Schulte : "The Integration Scenario: New Technology, New Standards and New Designs", Gartner Web Services & Appl. Integration Conf. 2003 (2003)
- (21) R. Davies : "Data Caching: A Prerequisite to the Enterprise Service Bus", Business Integration Journal, October 2003, pp.41-44 (2003)
- (22) OASIS : <http://www.oasis-open.org/>

田 中 哲 雄

(正員) 1961 年 11 月 3 日生。1985 年 3 月大阪大学基礎工学部情報工学科卒業。1987 年同大学大学院修士課程修了。同年(株)日立製作所入社。現在、同社システム開発研究所勤務。システム連携技術、企業間電子商取引等の研究開発に従事。情報処理学会、電子情報通信学会各会員。電気学会「企業情報システムにおける連携技術調査専門委員会」幹事。

湯 本 真 樹

(正員) 1971 年 4 月 6 日生。1995 年 3 月大阪大学工学部情報システム工学科卒業。1997 年同大学大学院工学研究科情報システム工学専攻修士課程修了。1998 年同専攻博士課程修了。現在、近畿大学理工学部電気電子工学科講師。情報処理学会などの会員。システム工学、知識情報処理の研究に従事。電気学会「企業情報システムにおける連携技術調査専門委員会」幹事。

齋 礼

(正員) 1961 年 10 月 3 日生。85 年 3 月早稲田大学理工学部機械工学科卒業。87 年同大学院理工学研究科修士課程修了。同年(株)日立製作所入社。2000 年 3 月大阪大学大学院工学研究科情報システム工学専攻博士後期課程修了。工博。2002 年 4 月広島国際大学社会環境科学部情報通信学科助教授、現在に至る。エキスパートシステムの研究に従事。人工知能学会などの会員。電気学会「企業情報システムにおける連携技術調査専門委員会」委員。