解 説 エレクトロニック・コマース

7. ECの技術動向：ソフトウェアのアーキテクチャ

Software Architecture for Electronic Commerce Systems by Tetsuya MASUISHI, Jun NITTA (Information Systems R & D Division, Hitachi Ltd.), Hiroshi FUKUOKA and Kikuo YOSHIMURA (Software Development Center, Hitachi Ltd.).

増石哲也[1] 新田 淳[1] 福岡 寛[2] 吉村紀久雄[2]

1 (株)日立製作所情報・通信開発本部
2 (株)日立製作所ソフトウェア開発本部

1. はじめに

本稿では，電子商取引（EC）を実現するためのソフトウェア・アーキテクチャについて述べる．

ECを実現するシステムと外部のシステムとの関係として，以下の2点をあげることができる．

（1） 世界各国をめぐるインターネットと接続して，不特定多数の顧客を相手に商取引を行う
（2） 既存の基幹システムと接続して，従来の基幹業務の一部として取り扱う

この2つの境界条件から，ECシステムを実現するソフトウェアに要求される観点として

（1） セキュリティ
（2） 種々のコンポーネントを接続するパラダイム

の2点をあげることができる．

1.1 セキュリティ

インターネット上での商取引を「安全に」行うニーズはきわめて高い．このため，ECを実現するためには，暗号[5]をかけて通信するというのが常識となっている．通信のトランスポート・レイヤで暗号をかけてしまうSSL（Secure Socket Layer）という方法がもっとも簡単である．しかし，決済などの商取引においては，買う人と売る人のほかに金融機関などが関係することが多く，SSLのような2者間の通信路での暗号化だけでは対応できないことが多い．また，暗号化・復号化の計算量・データ量のオーバーヘッドも考慮する必要があるので，必要なデータに必要なセキュリティ・レベルの暗号にするという細かいセキュリティが実現できない．

そこで，より細かいセキュリティを実現するためには，クレジットカード番号など特定のデータに強い暗号をかけることなどを実現するアプリケーション・レベルの通信プロトコルを用意する方式をとることが多い．この種の通信プロトコルの例として，Visa InternationalとMastercard Internationalとで制定したクレジットカード決済用のプロトコルSET（Secure Electronic Transaction）[6]がある．

通産省では，SECE（Secure Electronic Commerce Environment）として，SETを決済手順として包含し，かつ銀行決済などを統合した手順を制定しようとしている．2章では，SETを実現するためのソフトウェア・アーキテクチャの概要を説明する．

1.2 コンポーネントを接続する分散オブジェクト

ECを実用化するためには，インターネットをアクセスするためのクライアント・プログラム，インターネットからアクセスされる窓口に相当するWebサーバ[1]，電子商取引を実現するサーバ・プログラム，通常の業務を運営するための基幹システムのプログラムなどを接続して運用する必要がある．

これらの数多くの種類のコンポーネントを結合した大きなシステムを構築するためには，コンポーネント間のインタフェースを共通のパラダイムで表現することが望ましい．この種のパラダイムとしては，オブジェクト指向でコンポーネント間のインタフェースを定義する，分散オブジェクト（Distributed Object）が広く認められる方向にな

ってきている．分散オブジェクトのプラットフォームの仕様としては，業界主導の標準化団体OMG（Object Management Group）が規定しているCORBA（Common Object Request Broker Architecture）[3]と，MicrosoftのOLE（Object Link and Embedding）の下位レイヤにあたるDCOM（Distributed Component Model）[2]の2つが実用レベルになってきている．とくに最近は，この2つを接続するためのブリッジが製品として出されるようになってきている．

ECの場合，インターネット経由で商品を見て説明を読んで購入するという手順を踏むので，Webブラウザをユーザ・インタフェースとして利用する．Web系と分散オブジェクトをつなぐインタフェースがアーキテクチャ上重要な構成要素になる．これについて3章で述べる．

2. SET決済手順をもつソフトウェアの機能

2.1 SET決済手順の関与者

（1） カード所有者・商店・クレジットカード会社

SETは，クレジットカード決済のための決済プロトコルである．このため，カード所有者が商店に対してある金額の代金を支払う際，商店の背後に位置するクレジットカード会社がカード所有者の会員資格や与信限度額などの情報をチェックして，その支払金額についてクレジット決済を行っていいかどうかを判断する方式を採用している．このため，決済プロトコルとしては，カード所有者，商店，クレジットカード会社の3者間でやりとりするプロトコルが必要になる．

（2） 認証局

クレジットカードを利用して買い物をする場合，利用者がカード所有者本人であることを証明しなければならない．実世界においては店頭で，クレジットカードの裏側にあらかじめ記入したカード所有者の署名と，決済伝票の署名とを照合する手順が踏まれる．

ECの場合，カード所有者による署名は秘密鍵を用いて電子的に行われ，公開鍵を用いて電子署名の検証が行われる．この場合，第三者がこのカード所有者に成りすまして買い物をすることがないように，公開鍵とその所有者との対応関係を保証する手段が必要である．この手段として，対応関係を保証する認証局とそこで発行される認証書が用いられる．認証書は，実世界においては公的機関が発行する印鑑証明に相当する役割を果たす．

認証局は認証書の発行申請を受け付け，申請者に関する本人確認を行って，公開鍵が確かに申請者本人のものであることを証明する認証書を発行する．カード所有者は，商品購入前にあらかじめ認証局に自分の公開鍵に対する認証書を発行してもらう．SETでは認証書の申請から入手までの手順を電子的に行うためのプロトコルを規定している．

SETで用いられる認証書の形式は，X.509 V3に準拠しており，さらにSET用の拡張部分が仕様として規定されている．決済時にはこの認証書がカード所有者から販売店やクレジットカード会社に渡り，認証書に記載された公開鍵により署名が検定され，これによりカード所有者の本人確認が行われる．

SETでは，カード所有者のほか，商店，カード会社においても電子署名を行う場面があるため，これらの関与者が所有する公開鍵に対して認証書を発行するためのプロトコルも規定されている．またSETにおいては，複数の認証局がルート認証局をもとに認証関係による階層構造をなしており，上位認証局が下位認証局に対して認証書を発行する．

2.2 SET決済手順

カード所有者が購入する商品と数量とを決めると，カード所有者のクライアントマシンで購入申込書とクレジット申込書を作成し，それぞれ商店とカード会社の公開鍵で暗号化した封筒に入れて，商店に送付する．商店では，購入申込書の封筒を自分の秘密鍵で開け，クレジット申込書の封筒はそのままカード会社に転送する．カード会社では，クレジット申込書の封筒を自分の秘密鍵で開け，申込書に記述してあるカード番号の人物に対して，その決済金額をクレジット決済していいかどうかを判断して，YesかNoかの判断結果を暗号化して商店に返す（**図-1**）．

セキュリティの観点で，商店にはカード番号がわからない仕掛けになっている点に注意していただきたい．現実社会では店員にクレジットカードを渡すので，SETの方がよりきびしいセキュリティを実現しているとも言うことができる．

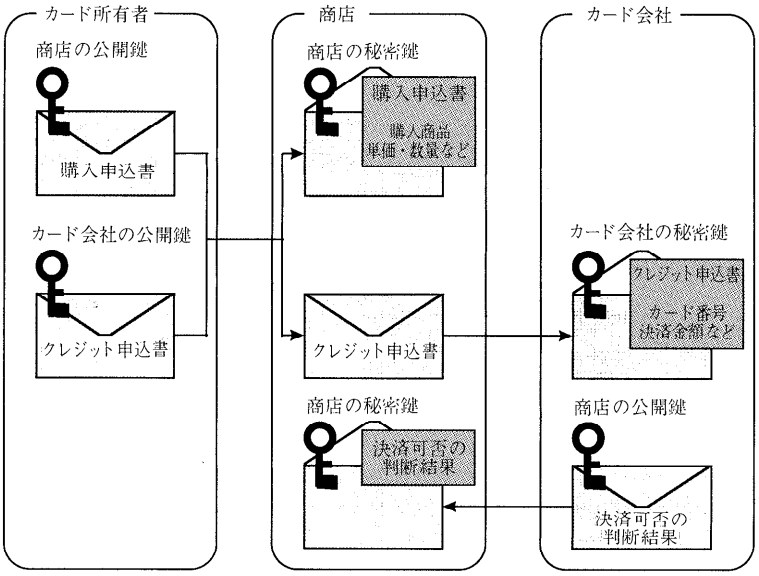


図-1 SET プロトコルの概要

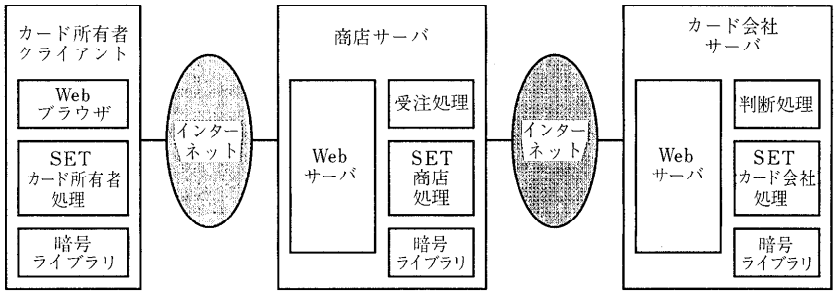


図-2 SET プロトコルを実現するソフトウェア構成の概要

2.3 ソフトウェアとしての実現方式

SET プロトコルを，ソフトウェアとして実装するためには，カード所有者クライアント，商店サーバ，カード会社サーバのマシンがインターネットで接続されている形態をとる必要がある．それぞれのマシンに最低限必要な機能は，**図-2** に示すとおり，SET プロトコルを処理する3者のプログラムのほかに，暗号ライブラリと，通信するための Web ブラウザと Web サーバから構成される．

図-2 に現れている構成要素のうち商店サーバの「受注処理」とカード会社サーバの「判断処理」は，商店およびカード会社の事務処理に依存した機能をもつ必要がある．それ以外の構成要素は，標準機能をもてばどの商店，カード会社にも適用可能なものと考えていい．

3. Web-ORB 連携アーキテクチャ

3.1 Web と ORB を接続する必要性

インターネットのユーザを対象とする EC にとって Web は不可欠なシステム要素である．Web サーバを窓口とする商店サーバのソフトウェアは，Web サーバのバックで動く形態となる．この際，Web サーバとのインタフェース（代表的なものとしては Common Gateway Interface）を経由してサーバ・ソフトウェアが呼び出されることになる．

サーバ・ソフトウェアの規模が大きくなると，これをモジュール化する必要性が生じる．EC のサーバ・ソフトウェアは，商店サーバの例だと，商品データベースとその表示系，買い物処理を取り扱うプログラム，決済プロトコルのうちの商店

が担当する部分，消費税・送料計算，受注データベースとその更新系，受注データの通信プログラムなど，さまざまな種類の機能から構成される．これらの機能を，きれいにモジュール化して実現するパラダイムが必要になる．

また，ECを実用化しようとすると，商店の受注処理やカード会社の判断処理において，メインフレームなどで構築された既存の基幹システムと連携させて動作させることが望ましい．クライアント・プログラム，サーバ・プログラム，基幹システムのプログラム，データベースなどいろいろなコンポーネントを同じパラダイムで表現できることが望まれるようになる．

この種の統一パラダイムとして，CORBAとDCOMに代表される「分散オブジェクト」をあげることができる（**図-3**）．CORBAとDCOMの分散オブジェクト呼出し機構を総称して，ORB（Object Request Broker）と呼ぶ．

3.2 Web-ORB連携方式

ここでは，WebとORBを用いて，ECソフトウェアを構成したソフトウェア・アーキテクチャを述べる．以下，ユーザの操作を追いながら説明する．

（1） ユーザが商品を眺める

ユーザがWebブラウザで商品を眺める際，ECソフトウェアとしては，**図-4**のように，商品データベースを検索して，動的にHTML文書を生成する機構（ページ生成機構）が必要になる．データベース機能を用いて，オンラインで商品の登録・価格更新などができる．ページ生成機構には，ページのレイアウトとして商品を配置するための穴の空いた状態を表現するページ・テンプレートを入力する．HTML上で穴の空いたところには，HTMLを拡張した言語により，データベ

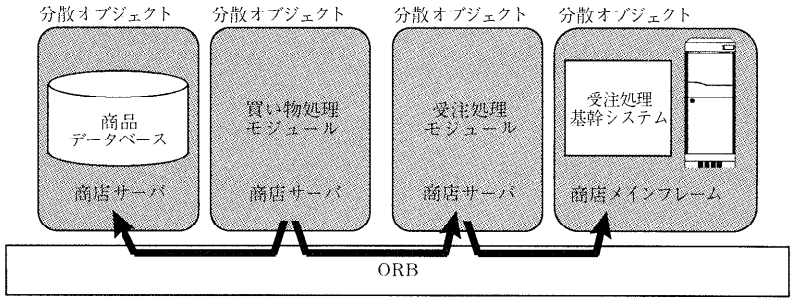


図-3 分散オブジェクトによるパラダイム統一

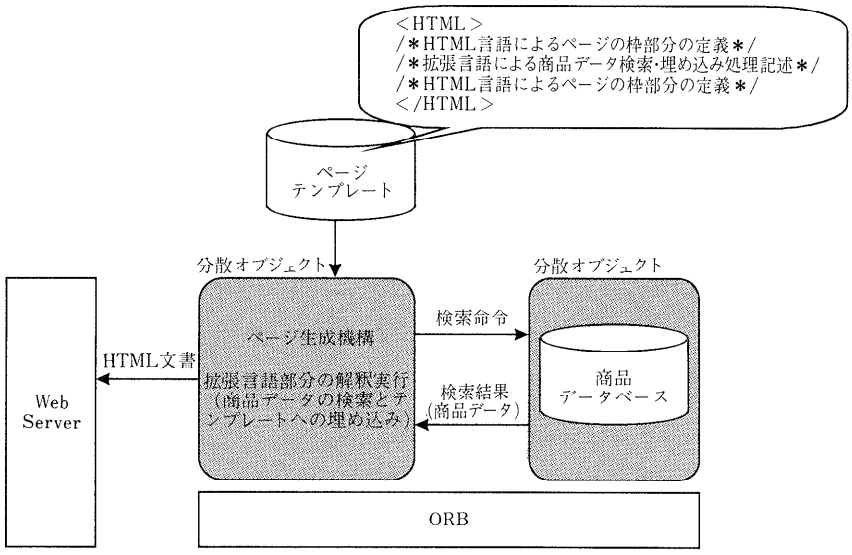


図-4 動的HTML文書表示

ース検索命令と，検索結果をHTML中に埋め込む命令が記述されている．ページ生成機構は，この拡張言語のインタプリタであり，これによって，ページ・テンプレートの穴を，商品データベースから検索した商品データで埋めることができるようになる．

ページ生成機構が解釈実行する言語の機能としては，基本的に分散オブジェクトのメソッドを呼出し，結果を受け取り，HTML中に展開する機能をもち合わせていればよい．したがって，ページ生成機構は，HTML中に書かれたスクリプトから分散オブジェクトを呼び出すゲートウェイであると解釈していいものである．

図-5には，さらに進んだ商品表示機能を示している．ここでは，コンテンツ選択機構というオブジェクトを設け，ユーザIDや過去の購入ログ，アクセスログから，表示する項目や内容をユーザにあわせて変更することを可能にしている．商店の会員などに特典を与えることや，ユーザの購入履歴に基づいて積極的に商品を提案することなどが可能になる．

（2） ページを次々にめくる

WebサーバとWebブラウザの間でHTML文書を転送するプロトコルHTTP（Hyper Text Transfer Protocol）は，HTML文書を転送するたびごとにコネクションが切られるという特徴がある．このため，ユーザがあるページを表示したまま何時間も放っておいても，ネットワークに負担がかからないという利点がある．しかし，この特徴は，あるページをアクセスしてきたユーザと次のページをめくったユーザが同じユーザであることを特定するためのセッション管理機構を要求することになる．技術的には，一般に「cookie」[7]と呼ばれるヘッダ情報を用いて，同じユーザかどうかを特定する方式をとることになる．図-4，図-5では簡単のためセッション管理機構を示していないが，**図-6**に示すようにWebサーバとページ生成機構との間に位置づけられるものである．

（3） 買い物をする

商品データベースから生成されたページから「購入する」あるいは「買い物かごに入れる」という操作を行うと，商品ID，購入数量などを記憶していく．

（4） お金を払う

購入商品に対して，代金を決済するとき動作するSETプロトコルは，アプリケーション・レベルでの暗号化を実現する．基本的な暗号化・復号化を行うライブラリは，いろいろなインターネット・アプリケーションに対して共通な機能を提供

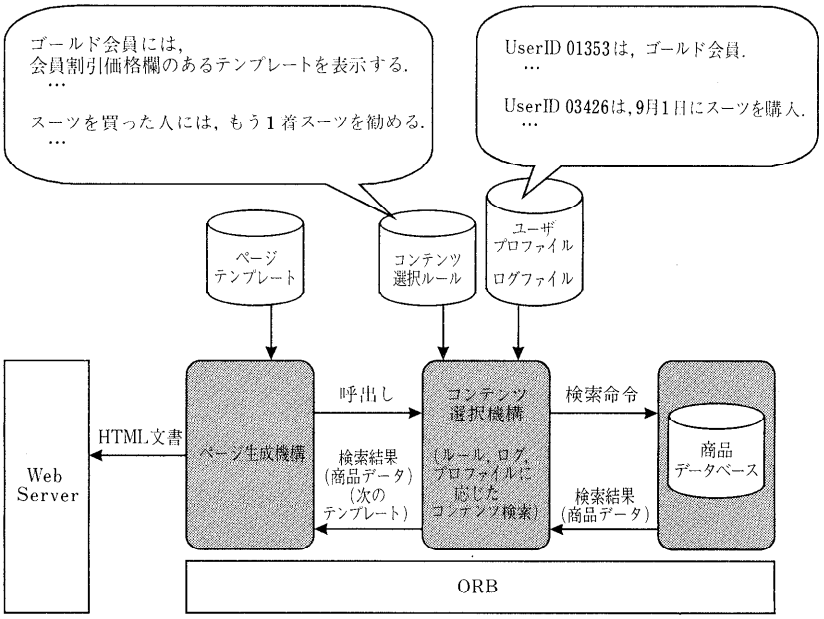


図-5 さらに進んだ動的HTML文書表示

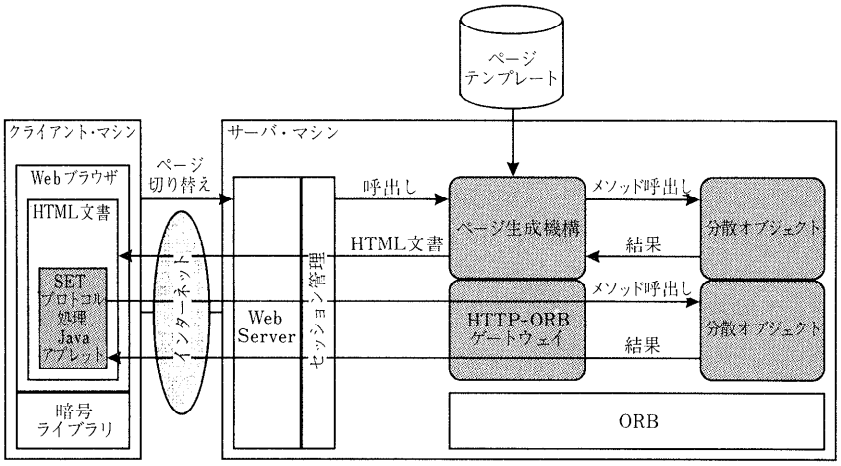


図-6 SET プロトコルを Java アプレットとして実現した場合

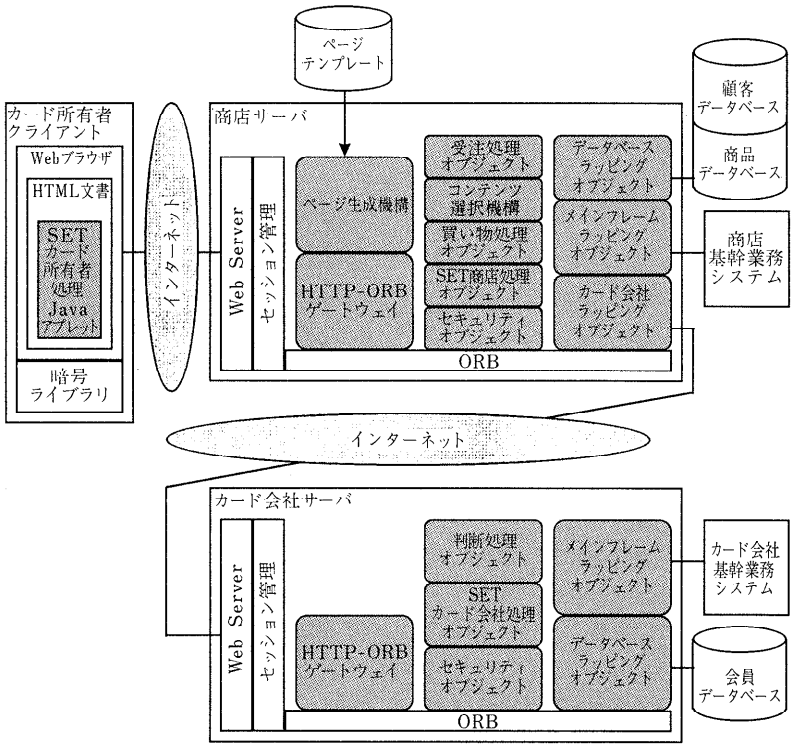


図-7 EC ソフトウェアのアーキテクチャの全体像

できるはずである．一方，SET プロトコルを処理するプログラムは，決済するその場で必要になるものである．決済機関によっては SET 以外のプロトコルが存在する可能性も考慮すると，SET プロトコル処理を行うプログラムは，できればクライアント・マシンに常駐させないほうが適当だと考える．

したがって，ソフトウェア・アーキテクチャとしては，図 6 に示すように，SET プロトコル処理プログラムを決済するページに乗せた Java アプレット[4]として実現し，暗号ライブラリはクライアントに常駐させてしまう方式が望ましいと考える．

この場合，Java アプレットから分散オブジェ

クトを呼び出す仕掛けが必要になる．SETがHTTP上に実現されるため，HTTPで通信されてきたデータをORBのオブジェクト呼び出しプロトコルに変換する機構，すなわち図-6でHTTP-ORBゲートウェイと示しているものが必要になる．これは，同じくゲートウェイであるページ生成機構と同列の位置づけにあり，JavaからRPC（Remote Procedure Call）のイメージで分散オブジェクトを呼び出せるような機能のものが望ましい．

3.3 全体像

今まで述べた構成要素に，カード会社サーバを加えると，**図-7**のようなアーキテクチャを描くことができる．ここで，カード会社ラッピングオブジェクトは，SET商店処理オブジェクトから呼び出され，Javaから分散オブジェクトを呼び出すのと同様のHTTPプロトコルで，カード会社のSETカード会社処理オブジェクトを呼び出す．このため，カード会社サーバには，HTTP-ORBゲートウェイが搭載されることになる．このほかにも，カード会社や商店における基幹システムをラッピングしたオブジェクトをサーバ内に配置している．

4. おわりに

ECではWebをフロントエンドとして，暗号化されたデータをHTTPで通信することを前提に進める必要があり，またWebサーバのうしろで稼動するサーバ・ソフトウェアには今後ORBが導入されるようになることが予想される．これを前提にソフトウェア・アーキテクチャを考えると，ここで述べたセッション管理，ページ生成機構，HTTP-ORBゲートウェイなどをWebサーバのうしろにもち，さまざまな機能が分散オブジェクトとして実現され，外部システムが分散オブジェクトとしてラッピングされた環境で実現される形態になるのは自然であると考える．

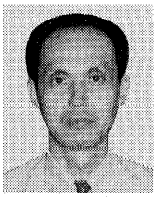
参考文献

1) Berners-Lee, T. et al.: The World-Wide Web, Comm. ACM, Vol. 37, No. 8 (Aug. 1994).
2) Box, D: Introducing Distributed COM and the New OLE Features in Windows NT 4.0, Microsoft Systems Journal (May 1996).
3) The Common Object Request Broker Architecture and Specification, Revision 2.0, Object Management Group (July 1995).
4) Java言語環境...A White Paper, 日本サン・マイクロシステムズ (July 1995).
5) Schneier, B: Applied Cryptography, Second Edition, John Wiley & Sons (1996).
6) Secure Electronic Transaction (SET) Specification, Version 1. 0, Mastercard International and Visa International (May 1997).
7) Spainhour, S and Quercia, V: WEBMASTER クイックリファレンス, 12章, オライリー・ジャパン (Apr. 1997).

（平成9年7月15日受付）

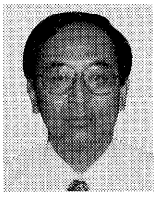
増石 哲也（正会員）

1958年生．1983年早稲田大学大学院理工学研究科電気工学専攻修士課程修了．同年(株)日立製作所入社．1990～1991年MIT Media Laboratory, Research Affiliate．現在，(株)日立製作所情報・通信開発本部主任研究員．入社以来，ソフトウェア開発環境の開発に従事．



新田 淳（正会員）

1958年生．1982年東京大学大学院理学系研究科物理学専攻修士課程修了．同年(株)日立製作所入社．現在，同所情報・通信開発本部主任研究員．入社以来，オンライントランザクション処理の研究開発に従事．



福岡 寛（正会員）

1953年生．1975年早稲田大学理工学部電気工学科卒業．同年(株)日立製作所入社．現在，同所ソフトウェア開発本部第2DC設計副部長．入社以来オンライン制御関連の開発に従事し，現在はECソフトウェア・プロダクト開発まとめ．

吉村紀久雄（正会員）

1948年生．1972年東京大学工学部計数工学科卒業．同年(株)日立製作所入社．現在，同所ソフトウェア開発本部AI設計部長．以来，OR・DSS・機械翻訳・AIやソフトウェア開発環境・EC関係のソフトウェア開発に従事．人工知能学会会員．