紹介

Java が実現する次世代コンピューティング —JavaCard によるエレクトロニック・コマースシステム—

瀧沢雅人, 塩田武臣*

[著者抄録] Sun Microsystems が開発した Java は単に新しいプログラミング言語ではなく、ネットワークコンピューティングを実現するキーワードでもある。また、常時携帯できる最小の Java 実行環境として JavaCard があり、このカードを使用すれば複数のプログラムを 1 枚のカードに安全に格納・実行することができる。CTC J-KIOSK 開発チームは、Java・JavaCard の特徴を取り入れたインターネット上での電子決済システムのデモを作成し、Java の実用性と可能性を証明した。

[著者付与キーワード] Sun, Java, インターネット, バーチャルマシン, スマートカード, JavaCard, EC(エレクトロニック・コマース), 電子決済, J-KIOSK

Introductory article

Next-Generation Computing realized by Java —Electronic Commerce System using JavaCard—

TAKIZAWA Masato, SHIOTA Takeomi

[Author Abstract] Java was developed by Sun Microsystems Inc., which is not only a new programming language but also a keyword that realize network computing. And JavaCard is placed as the smallest wearable environment to run Java. Once you take it, you can put several applications onto a card safely. Adopting Java and JavaCard advantages, we, CTC J-KIOSK Development Team has developed a Internet-Online Electronic Commerce demonstration systems. So we proved that Java is really practical and has a high possibility.

[Keywords by Author] Sun, Java, Internet, virtual machine, SmartCard, JavaCard, EC (Electronic Commerce), J-KIOSK

* 伊藤忠テクノサイエンス(株)西日本 IT ソリューション部
(〒532-0011 大阪市淀川区西中島6-1-1) Tel. 06(302)9943 E-mail: shiota@ctc-g.co.jp
Server Support grp., Western Region IT SOLUTION dept., ITOCHU TECHNO-SCIENCE Corporation
(1-1, Nishi-nakajima 6-chome, Yodogawa-ku, Osaka, 532-0011)

1. はじめに

Java が誕生して 3 年が経過した。インターネットの普及も手伝って、コンピュータに関係していなくても Java という言葉に聞き覚えがある人も多いのではないだろうか。近頃では一般の新聞やテレビでもしばしば取り上げられている。

Java を単なるプログラミング言語だと思っていた人は、最近のニュースについて不可解に思っているかもしれない。製造や金融などの話題の中に Java が登場しているのである。「VISA が電子マネー搭載 IC カードに JavaCard の採用を表明」、「Sony、デジタルビデオカメラに Java を採用」、「自動車制御電子部品にも Java Chip の搭載決定」、これらのトピックが示すように、机上のコンピュータ以外の環境にも、確かに Java は存在している。

デスクトップパソコンだけでなく家電製品や自動車などすべてのコンピュータが Java を理解できるようになることで、各機器どうしはネットワークを介してつながり、共通の言葉で話すことができるようになる。これが Java が目指すコンピュータ環境「Java Computing」である。

「Java Computing」の可能性に対しては、IBM や Microsoft といったハードウェア/ソフトウェアベンダはもちろんのこと、家電メーカ、金融、通信、交通、教育など、直接コンピュータに関わりのない業界からも注目が集まっている。Java Computing は、従来のコンピュータの利用形態を根底から覆す可能性さえも持っている。

この文章は Java Computing の可能性を探ることを目的として書かれた。まず Java の基本的な成り立ちと仕組みを紹介し(2 章)、次に最小の Java 実行環境である「JavaCard」について解説する(3 章)。そして最後に Java と JavaCard を用いて構築した電子決済デモシステム「JavaCard 電子マネーシステム」を例に、Java が実現する次世代コンピューティングについて論ずる(4 章)。

2. Java

Java は多方面から注目されているにもかかわらず、それ自身が多彩な顔を持っているため、あまり明確なイメージを持たれていない。この章では「Java Computing」の根本である Java の背景と成り立ちに触れ、その特徴を解説する。

2.1 Java の歴史

Java は 1995 年 5 月に Sun Microsystems より発表された。当初は家電製品向け組み込みデバイス用のプログラミング言語であった「Oak」が、インターネットの急激な発展を追い風に「Java」として再登場することとなる。

家電などに用いられている組み込み型のコンピュータでは、従来、独自の OS 上に独自のプログラムを作成していたが、生産性や再利用性が低いという問題点を抱えていた。この点に目をつけた Sun は 1990 年、プラットフォーム(コンピュータの種類やメーカなど、プログラムが実行される環境)に依存しないプログラミング言語の開発に乗り出した。そして「Oak」と呼ばれるコンパクトで高速、安全、さらにネットワーク対応といった特徴を備える言語が完成した。Sun ではこの Oak をもとに、対話式テレビの開発を手がけたが、この対話式テレビのプロジェクト自体は中止となってしまった。

一方、同時期のインターネットにおいては、それまでのテキストベースの情報のやり取りから、Mosaic といわれる最初のブラウザを皮切りにグラフィックベースの Web が急速に広がっており、それとともにインターネット自身が広く一般に浸透するようになってきていた。こうした状況の中、中断しかけていた Oak は、Web を中心としたインターネットに適用させるという変更を加えられて生まれ変わることとなった。Web 上で実行可能なプログラミング言語として登場した Java は、1995 年の Sun World'95 というショーにおいて、Sun から正式な言語として発表されることとなる。Java がインターネットを考慮しながら作ら

れたことは注目すべきことであり、ネットワークとの親和性が高いといわれるゆえんでもある。

現在、Netscape 社の Netscape Navigator や Microsoft 社の Internet Explorer といった代表的な Web ブラウザも当然 Java をサポートしており、Java はすでにインターネットに広く浸透している。当初家電製品向けの組み込みデバイス用言語としてスタートした Java は、こうしてインターネット上で成功を収めることとなったが、Sun はこの成功を「Java Computing」実現のための第1ステップにすぎないと見ている。Sun が掲げる「Java Computing」というキーワードは、デスクトップ上のコンピュータだけでなく、当初の目的であった家電製品をはじめ、携帯電話・PDA(携帯端末)・自動車などすべてのコンピュータに Java 環境を実現することを意味している。

2. 2 Java の仕組み

Java の特徴を簡単に列記すると「アーキテクチャ非依存」「ネットワーク対応」「セキュリティ」「オブジェクト指向」といった言葉が並ぶ。

Java はソース、バイナリいずれにおいてもアーキテクチャ非依存である。つまりあるソースコードがあって、一度それをコンパイルしてオブジェクトコードを作成すれば、あとは再コンパイルすることなしに Sun Solaris はもちろん Windows95 や Windows NT, Macintosh など様々なマシンの OS 上で同じプログラムを動作させることができる (Sun はこれを Write Once, Run Anywhere と称している)。

C や C++ などの通常のプログラミング言語では、ソースコードをコンパイルすると、その環境に特化した機械語プログラムが生成されるが、Java ではコンパイルによって機械語ではなくバイトコードと呼ばれる中間的なオブジェクトコードが生成される。バイトコードはアーキテクチャ (プログラムが実行されるコンピュータや OS の種類) からは中立で、実行環境の差異は、アーキテクチャごとに実装されたインタプリタ (翻訳機) を含む仮想コンピュータ (=バーチャルマシン、以下

図1 Java の動作環境

VM)によって吸収されており、バイトコードから機械語への最終的な変換は、実行時に VM 内のインタプリタによって動的に行われる (図1)。

Java にはアプリケーションとアプレットという二つの実行形態が存在する。アプリケーションは、自分のマシンのプログラムを自分のマシン上で動かす一般的な実行形態であるが、アプレットはネットワークからダウンロードしてきたプログラムの実行を前提としている。たとえば Java 対応の Web ブラウザでは、サーバにある HTML 文書を取ってきて表示するだけでなく、Java アプレットをダウンロードしてブラウザ上でそれを実行させることができる。Web ブラウザは Java 対応によって、単なる情報閲覧端末からネットワーク対応の一つの仮想コンピュータの機能を持ったといえる。

しかし、ここで懸念されるのが、セキュリティの問題である。インターネットを通じて世界中の Web サーバからプログラムをダウンロードし、ブラウザ上で実行できることは便利である反面、悪意をもって作られたプログラムによって、マシン上の個人情報を盗まれたり、コンピュータウイルスに感染してしまうおそれがある。これに対して Java では「Sand Box (砂場)」という概念を取り入れている。ネットワーク経由で取ってきたアプ

図2 Java アプレットの動作概念図

レットは、実行マシンの資源（ファイル・プリンタ・ネットワーク etc.）へのアクセスを制限されるというもので、アプレットは砂場（=Web ブラウザなどの仮想マシン）の中だけでしか自由に動けない。ただ、砂場内だけでは印刷やファイル保存などが全くできなくなってしまうため、作成者がアプレットに施した電子署名をチェックして、いくつかの権限をアプレットに与えるような仕組みが採用されている。また VM は、プログラム実行前にバイトコードの妥当性を検証しており、不正なバイトコードの実行を防いでいる。こうしたセキュリティへの対応がなされていることは、インターネットでの利用を考えると必要不可欠な機能である（図2）。

2.3 プログラミング言語としての Java

Java はオブジェクト指向の高級言語である。開発環境である JDK には、入出力やネットワーク機能、GUI 部品などの基本的なライブラリが含まれており、これらのパッケージを使えば、イベント処理や通信プロトコルなど低レベルの処理に関与することなく、多機能アプリケーションを作成することができる。さらにデータベースアクセスや Web ブラウザ機能、メール送受信などの処理を行うパッケージが次々とリリースされており、これらによって Java が実現できるアプリケー

ションの幅はさらに大きく広がっている。

Java のプログラムのうち、再利用を前提としたモジュールの仕様が規定されており、これを「Java Beans」と呼ぶ。Java Beans は、Java によって構築されたソフトウェア部品で、GUI ベースの開発ツールを使えば、これを視覚的にカスタマイズしたり、部品同士を組み立てることが可能である。Java Beans を組み合わせることで、開発者は1行もコードを書くことなくアプリケーションを作成することすらできる。ビジネス向けの Java Beans としては、「Enterprise Java Beans」が今年の春に発表されており、本格的なビジネス・アプリケーションの構築を支援する。

Java はこの他にもいくつかの言語的特徴を持っている。まず言語仕様がきちんと規定されているため、他言語にありがちな型のサイズの違いなどを考慮しなくてよい。また、メモリは実行時に自動的に割り当てられ、直接的なメモリアクセスはできない仕様になっている。反対に使用されなくなったメモリは自動で開放されるという点も大きな特徴である。この自動メモリ管理はガベージコレクション（=ごみ集め）と呼ばれているが、これによりプログラマは、最も発見しにくく、デバッグに時間がかかるメモリリーク（プログラムがメモリの領域を開放せずに資源を浪費してしまうこと）から解放されるようになった。

このほかにも Java の新しい技術が日々開発され、発展している。Swing と呼ばれる GUI 部品パッケージはより豊かな GUI 環境を提供している。Java の欠点といわれる実行速度の遅さは JIT（ジャスト・イン・タイム）コンパイラや Hot Spot 技術などによって解消しつつある。さらに2D 画像や3D 画像の処理を行う Java 2D/Java 3D パッケージ、動画や音声などのマルチメディアデータの処理を行う Java Media Framework など、枚挙にいとまがない。

2.4 広がる Java Computing の世界

現在の Java のメジャーバージョンは JDK 1.1 で、まもなく JDK 1.2 がリリース予定になっている

る。バージョンアップに従い Java はさらに高機能化が進んでいるが、デスクトップマシン以外の機器へ現行の Java を適用するには明らかにオーバースペックである。たとえば高度なグラフィック機能は家電製品などの組み込み用機器には必要ない。またこれらの機器ではメモリの容量も多くないため、フルセットの Java の搭載は不可能である。

そこで、Java のサブセット版(縮小版)として、携帯電話・PDA 向けの Personal Java、家電等の組み込み機器向けの Embedded Java、スマートカード向けの Java Card が定義された。こうした環境においても、クロスプラットフォーム、安全、コンパクト、ネットワーク対応といった Java の特徴が有効になる。開発者は Java を利用すれば、共通の開発環境やアプリケーションの再利用が可能になり、開発効率を格段に高めることが可能になる(表1)。

次章では、これらの Java のサブセット版のう

ち、常時携帯することが可能な最小の Java 環境である JavaCard に焦点を当てて解説する。

3. JavaCard

現代社会はカード社会といわれるものの、日本では長い間磁気カードが主流であり、それゆえ、いまだに変造・偽造のトラブルが多発している。

この章では、磁気カードの現状に続いて、情報量・安全性の点で磁気カードに優るスマートカードについて述べ、さらにマルチ機能カードを実現する新しいスマートカードの標準「JavaCard」について解説する。

3.1 現代カード事情

私たちの財布の中には様々なカードが混在しているが、それらは機能的に下記のような分類ができる。

・決済・プリペイド系

表1 Java の実行環境

種類	位置づけと要求仕様	適用範囲
Java	通常の Java(エンタープライズ Java)	デスクトップコンピュータ
Personal Java	ディスプレイ・入力装置を持つ装置向けの Java ・プロセッサ: 32bit CPU/クロック 50MHz 以上 ・ROM: 2MB 以上 ・RAM: 最低 512KB~1MB ・ネットワーク接続が可能である ・キーボードかそれに代わる入力装置を備えていること(タッチパネル・リモコンなど) ・ほとんどのアプレットの実行が可能であること ・主にリアルタイム OS 上で稼動	家電製品・携帯電話・PDA(携帯端末)・セットトップボックス(インターネット TV)
Embedded Java	組み込み用デバイス向け Java ・リアルタイム OS 上で稼動 ・メモリ管理をサポート ・スレッド管理をサポート ・ファイル入出力など(オプション)	家電製品・携帯電話・ポケベル・FAX・ビデオ・ルータ
JavaCard	最小の Java 環境 ・ROM: 16KB 以上 ・EEPROM: 8KB ・RAM: 256B 以下の機能はサポート外 ・ガーベッジ・コレクション ・マルチスレッド ・セキュリティ・マネージャ	スマートカード・リングや時計への組み込み

現金の支払いの代用機能や、現金と同様にそれ自体が価値を持つカード

(例)テレホンカード・プリペイドカード・クレジットカード・キャッシュカード

・認証系

自分の資格・所属を証明するためのカード

(例)免許証・社員証・会員証

・ロイヤリティ系

何らかの特典を受けるまでポイントを蓄積するカード

(例)スタンプカード・割引券

カードに情報を保持させる技術としては、印刷やスタンプなどの手段のほか、磁気カードが広く普及している。磁気カードは1枚あたりの製造コストが低いため、テレホンカード・クレジットカード・キャッシュカードなど身近なカードとしても一般に広く浸透している。

3. 2 磁気カードからスマートカードへ

磁気カード内のデータは、磁性体の持つ磁場の強弱で記録されている。このストライプが2進数の0・1に対応しており、データ読み取り時はこの0・1の羅列を再び意味のあるデータとして復元しているだけである。そのためカードの複製・データの改ざんは比較的容易であると言わざるを得ない。プリペイド系のカード、とりわけテレホンカードなどの偽造はサービス規模が大きいだけに被害もまた甚大である。磁気カード自体に金銭的価値を持たせることは特に被害拡大のおそれを持つため、システム上で利用額の上限をやむなく下げなければならない場合が多い。

磁気カードでは記録容量の問題だけでなく、こ

うしたセキュリティの低さから、重要なデータをカードに持たせることが不可能であった。そのため重要データは、カードでなくサービス提供者が集中して管理しなければならなかった。これに対してデータをカード内に保持しながらも、「必要なデータのみ」を「安全に」カードから出し入れできるようにしたのがスマートカードである。

スマートカードはICチップがカードに埋め込まれたもので、保持できる情報量は磁気カードの100倍～1,000倍にもなる。またデータのやり取りは決められた手順を踏まなければ読み込みさえできない仕組みとなっているため、カード内のデータを高いセキュリティで守ることが可能となっている。フランスでは、1983年から公衆電話のテレホンカードにスマートカードが早くも採用されている。スマートカード型テレカの発行コストは若干高いものの、日本の磁気カード型テレカの偽造の被害を考えれば賢明な選択であったといえる。こうした状況の中、日本でも1999年度からスマートカード型テレホンカードの発売が始まり、公衆電話も順次これに対応していくことが決定している。

スマートカードと外部の読取装置とのインタフェースには、物理的な接点を持つ「接触式」と、接点を持たない「非接触式」の2通りの形態があり、「非接触式」は通信可能距離によってさらにいくつかに分類される。来年NTTから登場する新しいテレホンカードは、10cm以内の距離で読取可能な「非接触式(近接型)」である。また最近では、「接触式」「非接触式」両方の機能を組み合わせた「コンビ式」カードも登場している(表2、図3)。

表2 スマートカードの分類

読取装置との距離		読取装置までのアクセス距離	通信方法
接触式		0	電流
非接触式	密着型	～数 mm	電磁誘導(4.9152MHz)
	近接型	～10cm	電磁誘導(13.56MHz)
	近傍型	～70cm	電磁誘導(13.56MHz)
	遠隔型	～数 m	マイクロ波方式

ICカード化されたフランスのテレホンカード。実際のICチップはこの端子の下に隠れており、その基盤の大きさはわずか5mm四方である。表面の8個に分かれた端子はICチップと外界とのデータの入出力のインタフェースとなっている。

図3 接触式スマートカード

3.3 「考える」カードスマートカード

スマートカードのうち、CPUを持たず、データを読み書きする機能のみを持つカードをICメモリカードと呼んでスマートカードと区別する場合もあるが、通常のスマートカードはCPUを持ち、カード内部でプログラムを実行することが可能である。つまり、スマートカードは一つの独立した小さなコンピュータであるといえる。カード内に格納されたデータはCPU上で走るプログラムによって管理されており、これによってデータの入出力を制限し、単なるメモリカードよりもより高度なセキュリティ機能を確保している。

スマートカードは、海外ではすでに電子マネーや携帯電話の認証カードとして実際に使用されており、日本でも一部の会社の社員証や入退室カードなどに採用されている。また、現在各地で行われている電子マネーの実証実験も、そのほとんどがスマートカードを用いたシステムである(図4)。

カードの内部で任意のプログラムが実行できるため、データのアクセスに制限をかけたり、重要データを暗号化してからカード外に出すことが可能となる。

図4 スマートカードの動作原理

3.4 スマートカードの標準化

ICカードメーカーは自社カード向けに独自のOS・プログラムを実装しているため、アプリケーション開発は各ICカードメーカー依存とならざるを得ず、また、開発したアプリケーションは当然

別メーカーの製造するカード内には入れることができなかった。さらに複数の機能を独立して持たせるようにするためには、セキュリティを考慮したプログラムを記述し、デバッグを慎重に行って複数のアプリケーションが独立できるよう配慮しなければならなかった。

こうした状況では、どうしてもカード発行のコストや時間がかかってしまい、スマートカードのメリットが生かしきれない。

そこでメーカーに依存しないカード内アプリケーションの開発ができることと、1枚のカードに入れた複数のアプリケーション間のセキュリティを保てる実行環境が、スマートカードの標準化の要件として求められてきた。

3. 5 JavaCard の登場

2章で述べたように、Javaはプラットフォームに依存しないプログラミング言語であり、またアプレットという制限された実行形態が存在するため、実行されるアプリケーションが他のアプリケーションの実行やメモリ領域に侵入することができない。

この特性をカード内にも応用し、スマートカードの標準として名乗りを挙げたのが「JavaCard」である。1996年10月に「Java Card API v1.0」が、1997年10月には「Java Card API v2.0」の仕様がリリースされ、現在では世界の主要なカードメー

カのほとんどがJavaCard採用を表明しており、数社から実際のJavaCardが発売されている(図5)。

JavaCardは、従来のスマートカードに加えて次の三つの特徴を備えている。

① 標準化されたオープンな内部仕様

カード内のプログラムがJava言語で記述できるため、アプリケーション開発はJavaプログラマなら誰でも可能となった。これにより開発者の確保が容易となった。

② カードに非依存のアプリケーション開発

JavaCardという統一された環境を前提にアプリケーションを開発することで、複数のメーカーのカード上で同じプログラムを実行させることができるようになり、アプリケーション開発期間の短縮・プログラムの再利用性の向上が実現した。

③ マルチアプリケーションの実行環境

複数のプログラム(Javaアプレット)を別個に追加・更新できることで、1枚のカードへの複数サービスの相乗りが可能になり、発行コストを分散させることができる。

JavaCardの仕様である「Java Card API」を実装したデバイスはカードにとどまらず、指輪や時計にもおよんでいる。これらはカードとともに、携帯できる最小のコンピュータとして様々なシーンでの活用が期待される。

図5 JavaCard (Schlumberger社のCyberflex)

図6 JavaCardの動作原理

図7 JavaCard Acceptance Vision

3. 6 JavaCard が描く近未来

JavaCard のマルチアプリケーション機能は、1枚のカードで全く別個のサービスを提供することを可能とした。比較的高いカード単価も複数の発行主でコストを分担することによって低減される。財布にあふれるカードや硬貨を1枚のJavaCardに集約することも可能である。

図7はJavaCardの今後の展開を予測したものである。JavaCardは従来のカードに取って代わるだけでなく、カードが使われなかった新しい分野にも適用が可能である。

コンビニエンスストアに行って、何も入っていない空のJavaCardを購入し、自分に必要なサービスをカードに順次追加していく…といった仕組みが数年後に本当に実現しているかもしれない。

4. JavaCard による EC システムの構築

本章では、EC、特にインターネット上での少額電子決済の現状を述べたあと、CTC大阪支店で開発されたJava, JavaCardによる電子決済デモ「JavaCard電子マネーシステム」を紹介する。

4. 1 EC の現状と動向

EC (Electronic Commerce: 電子商取引) と一口にいても、企業間の取引の決済から、個人対商店の少額の支払いまで様々な形態があるが、ここでは電子決済の形態の一つである電子通貨に焦点を当てて話を進める。

財布の代わりに、持ち歩き可能なスマートカード内に電子現金を格納し、現実の店舗での支払い

に対応するものとして、「MONDEX マネー」や「VISA キャッシュ」がすでに実用化あるいは実証段階に入っている。ともに財布に代わる少額決済ツールとして機能し、街角に設置された専用のリロード端末からカード内にお金を引き落として再利用することができる。

一方、インターネット上での決済の方法としては、クレジットカードによる支払いが一般的だが、このほかに「BitCash」「First Virtual」「NetCash」「CyberCash」などがある。これらは、残高情報を中央のサーバで一括管理し、すべてのトランザクション(支払い手続き)をこのサーバが監視するものである。

4. 2 新しい電子マネーシステム構築の要件

しかし、手持ちのカードに貨幣価値を持たせながら、なおかつそれをインターネットを介した電子マネーの引き落とし・支払いに使用できるようなシステムはいまだ存在していない。

両者を組み合わせたシステム、つまり現実世界・仮想世界の両方での支払いに利用できる電子現金の仕組みが望まれるが、そのためにはいくつかの問題点をクリアしなければならない。

(a) カードとリーダ(読取装置)の問題点

「MONDEX マネー」と「VISA キャッシュ」は、ともに専用装置を使って電子貨幣の入出金を行っているが、インターネット経由の取引を行うためには、コンピュータと接続できる一般のICカードリーダでカードと通信を行わなければならない。

(b) 盗み見・改ざんの問題点

カードから出た取引メッセージは、カード所有者あるいは第三者の悪意によって、内容を盗み見られたり書き換えられるかもしれない。そこでカードから出力される前にメッセージを暗号化する必要がある。

(c) なりすましの問題点

「なりすまし」はインターネット全般の問題点だが、この場合はたとえば悪意のある者が別の人物になりすまして、電子マネーの支払いを横取り

すること等を指す。これを防ぐためには、自分が支払いをしようとしている対象が電子証明書を取得しているか確認する必要がある。

電子証明書は、自分が何者であるかを信頼できる第三者に証明してもらうもので、証明書は Verisign などの認証機関から交付を受ける。証明を受けた者は自分の WWW ページや Java アプレットに電子署名を施し、そのコンテンツの作者が自分であることを保証する。企業が電子証明書を取得するためには、登記簿謄本や印鑑証明書の提出などを行わなければならない。

(d) トランザクションの問題点

どれだけ上記のセキュリティ面での対策を施したとしても、インターネットのネットワーク特性上、通信の断絶などが発生することを考慮しなければならない。特に支払いの手続き中は、支払い者と受け取り者の金額に不整合が生じた場合の対策を考慮しなければならない。

次節で紹介する「JavaCard 電子マネーシステム」の開発は、こうした問題に配慮しながら、新しい電子マネーシステムの一例を提示すべく行われた。

4. 3 「CTC J-KIOSK : JavaCard 電子マネーシステム」の開発

現在、CTC 大阪支店のショールームでは、Java の最新テクノロジーを紹介する目的で「CTC J-KIOSK」コーナーを設置している。ここでは、Java の新しい技術を使ったデモンストラシンプログラムを展示しており、その第1弾が以下に紹介する「concept1-JavaCard 電子マネーシステム」である。

「JavaCard 電子マネーシステム」は、将来のスマートカード社会に実現すると思われる電子決済システムを予測し、実際のデモ化を Java および JavaCard を用いて行ったものである。本システムは、カードに電子現金を格納しながらも、インターネット経由でその支払いを可能にしているのが特徴である。

いまのところ、カードと通信するための IC

カードリーダ自体がまだ普及していないのが現状であるが、数年後にはリーダ装置がパソコンにも標準搭載され、街角にもスマートカードが使用可能な公衆キオスク端末が登場するだろうという予測もある。

本システムもまた、家庭や商店だけでなく、街頭、オフィス、駅や図書館など、様々な場所でサービスが提供されることを想定して作成されている。現在のデモでは以下のサービスが受けられるようになっている（図8）。

- ① JavaCard 内の電子マネーの残高照会
- ② インターネット銀行での JavaCard への預金引き出し
- ③ インターネット商店での買い物と JavaCard による決済

4. 4 「CTC J-KIOSK : JavaCard 電子マネーシステム」で使われている Java 技術

利用者は残高表示アプリケーションでカード内の残高を確認しながら、WWW ブラウザでイン

図8 JavaCard 電子マネーシステム

図9 JavaCard 電子マネーシステムで使われている Java 技術

ターネット銀行やインターネット商店を訪れ、電子マネーの引き出しやオンラインショッピングを行うことが可能となっている。このシステムを実現するために、いくつかの Java の新技術が使用されている。

たとえば、別のマシン上で動く Java のプログラム同士を通信可能にする「Java RMI」は、通信の下位プロトコル(データを流すための基本的な取り決め)を意識することなく、Java のみを使ってお互いのメソッド(手続き)を呼び合うことができる。また、Java オブジェクトからデータベースへのアクセスを実現する「JDBC」や、アプレットの認証技術、カード内で実行される Java のサブセット(縮小版)である「Java Card API」(3章参照)などがシステムの各部で使用されている(図9)。

5. お わ り に

「JavaCard 電子マネーシステム」の開発者3名はいずれも Java の初心者であったが、約11か月の

研修ののち、アプリケーションの開発をスタートし、3か月でプロトタイプを完成することができた。これは、Java が習得しやすく使い勝手のよい言語であることを示唆している。また、デスクトップ上の通常の Java のプログラミング手法が、JavaCard 内で動作するアプリケーションの開発にもほぼ同様に適用できるため、ここでも開発の効率を高めることが可能であった。

今後の展開としては、JavaCard のマルチアプリケーション実行のメリットをアピールするため、図書検索システムのような別システムを構築し、1枚のカードで複数のサービスが受けられるようにする予定である。

Java およびすべての Java 関連の商標およびロゴは、米国および他の国における米国 Sun Microsystems, Inc の商標または登録商標です。Sun, Sun Microsystems は、米国および他の国における米国 Sun Microsystems, Inc の商標または登録商標です。その他の各企業名およびその製品名は、各企業の商標および登録商標です。

~~~~~ 参 考 文 献 ~~~~~

- 1) 山内秀樹. Java が切り拓く近未来—Java Card. New Wave. vol.26 p.6-7(1998) CTC ユーザ会
- 2) Rankl Effing. SMART CARD HANDBOOK. WILEY, 1997
- 3) Peter Wayner 著, 川副博監訳. デジタルキャッシュテクノロジー. ソフトバンク, 1997
- 4) 日経デジタルマネー・システム編. デジタルマネーのすべて. 日経 BP 社, 1997
- 5) 月刊カードウェーブ. 5-10月号(1998)シーメ
- ディア
- 6) Schlumberger. JavaCard「Cyberflex」について(<http://www.slb.com/>)
- 7) Gemplus. JavaCard「GemXpresso」について(<http://www.gemplus.com/>)
- 8) Sun. Java 関連情報ページ(<http://java.sun.com/>)
- 9) Java for Smart Card. 月刊ジャバワールド. 4-8月号(1998) IDC コミュニケーションズ