特集

携帯電話もECサイトも続々と採用！

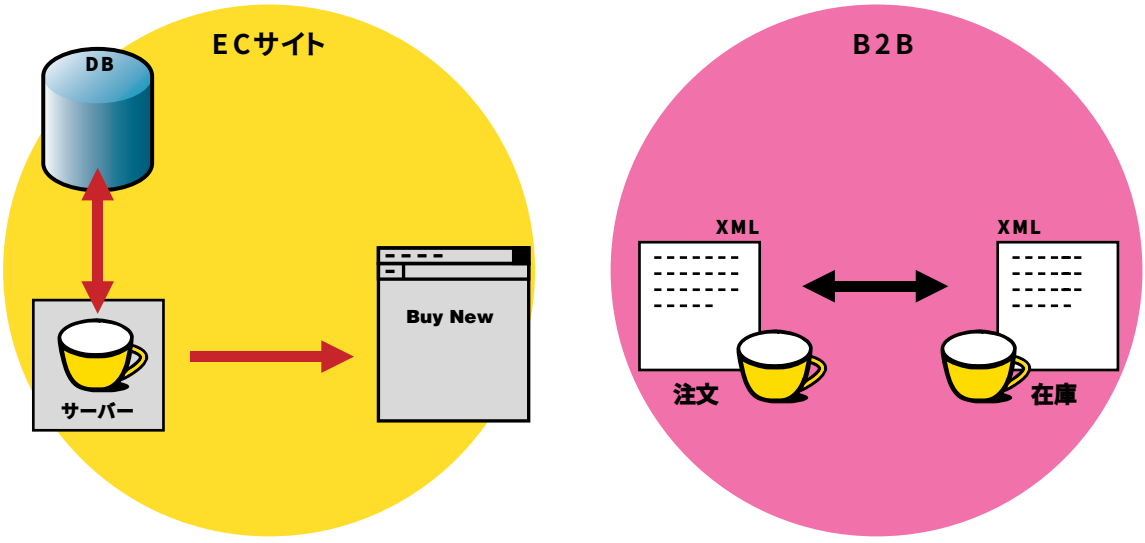
All about Java2

最新 Javaのすべて

今、Javaが熱い。
ASPのようなサーバー上のアプリケーションが社会の基盤となり、
携帯電話などのあらゆる機器がインターネットに接続できるようになった現在、
Javaがようやくその真価を見せ始めた。
この特集ではその現状をレポートしよう。
インターネット上でビジネスを展開していくうえで、
これから何の技術に注目すればいいのか探している人には、
Java以外に答えはない。

株式会社リンコム 清水 航＋編集部

Cover Photo : Nakanura Tohru
Photo : Nabeshima Akiko



世界がJavaに大注目!

「一度書いたらどこでも動く」と言われながら、何ができるのかはなかなか見えてこなかったJava。毎年次々と新しい機能が発表され、実態がわかりづらくなってきたが、最近ようやくその実力にふさわしい方向性が見えてきたのではないだろうか。まずはJavaのこれまでと現状についておさらいをしてみよう。

Java＝アプレットはもう古い!

いまだに「Java＝アプレット」と考えているなら、それはもう古い。今世界的に注目を集めているJavaは、パソコンのブラウザー上で動くアニメーションではない。eコマースのサービスを提供するサーバーの上で動くJavaと携帯電話などパソコン以外の情報機器の上で動くJavaだ。もともとJavaはアプレットのためだけの言語ではない。当初から、本格的なアプリケーションも開発できる汎用的な言語として作られているのだ。しかも、開発効率の高さ、ネットワーク環境を前提としていること、高度なセキュリティー、そして実行環境さえあればどんな機器の上でも動くという特徴を備えた先進的な言語だ。もう「ブラウザー戦争」のようにパソコンで動くソフトの機能の高さばかりが注目される時代ではない。パッケージソフトからアプリケーションサービスへ、パソコンから情報家電へ時代の流れが変わってきた今、ようやくJavaが活躍するのにふさわしい環境が整ってきたのだ。

ECサイトにJavaは常識

1999年のJavaOneでJavaは大きな転換点を迎えた。Javaが大きく3つのカテゴリー、つまりJ2EE、J2SE、J2MEに分けられたことだ（P.206～207参照）。これと前後して、Javaをサポートしたアプリケーションサーバーが各社から続々と発表されてきた。さまざまなOSが存在し、また24時間365日サービスを止めることが許されないサーバー分野では、「アプリケーションを開発するならJava」という認識が広まってきたのだ。

サーバー上でアプリケーションを動かし、WWWを通して問い合わせやその結果をやり取りすれば、クライアントはどんな機器でもかまわない。シンプルなWWWブラウザーさえあればいい。そんな具合に、現在コンピュータを使う目的は、eコマースのようなインターネット上のサービスを利用することが中心になっている。こうした時代に対応するには、サーバー上で複雑なアプリケーションを開発するのに適した環境が必要だ。Perlを使ったCGIではパフォーマンスや開発効率の点で力不足。安心してビジネスに使える環境は今のところJavaしかない。

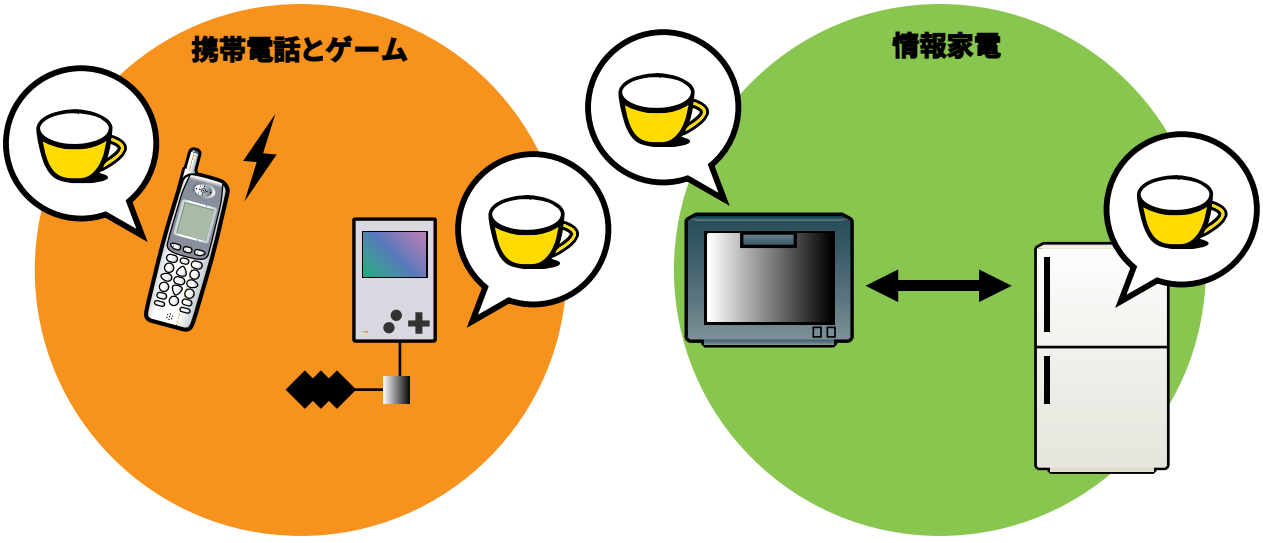
J2EE（Java 2 Enterprise Edition）として仕様が徐々に固まってきたことで、サーバー上のJavaは本格的な普及期を迎えている。Amazon.comのようなECサイトをはじめとして、世界中で銀行や航空会社などがJavaをサポートしたアプリケーションサーバーを採用している。

B2BにはXML&Java

今Javaを巡るもっともホットな話題は、JavaとXMLの組み合わせだ。しかも、企業間（B2B）の電子商取引との関係で語られることが多い。それはなぜだろうか。

XMLは、あらゆるデータを標準的な方式で記述するためのマークアップ言語だ。複数のコンピュータが別々のフォーマットを使っている場合に困難だったデータ交換が容易になる。一対一の企業同士ならデータ形式を揃えることも簡単だが、電子商取引が不特定の企業同士や異なる業界の間でデータを交換し合うeマーケットプレイスに発展してくると、

アプリケーションサーバー：CGIプログラムやJavaプログラムなどのおもにブラウザー上で動作するアプリケーションを配信するためのサーバー。ECサイトなどでWWWサーバーの負荷を軽減するために用いられることが多い。



XMLへの期待がますます高まることになる。そこではコンピュータ環境やデータ形式を統一することはもはや不可能になるからだ。各企業が持っているデータをいったんXMLに変換し、HTTPなどの標準的なプロトコルを通してやり取りするという方式が取られることになるだろう。

このようにXMLはインターネット時代の共通語として作られたものだが、XMLで書かれたデータはそれ自体は何の働きもしない。XMLを活用するには、同じようにインターネットに最適なプログラミング言語が必要になるのだ。そこでJavaが注目される。データはすべてXMLでやり取りし、XMLを利用するときはJavaを使う。これが次世代のスタンダードになる。

iモードにJavaが載る

Javaはもともと家庭用の電子機器を制御するために開発された。「携帯電話にJavaが載る」という話を聞いてもピンと来ないかもしれないが、出発点から見ると情報家電にJavaが組み込まれていくのは当然の流れだ。ウィンドウズとマッキントッシュがほとんどを占めるパソコンの世界とは違い、情報家電では、ゲーム機やオーディオ機器から、冷蔵庫などの白物家電、PDAや携帯電話のように環境はさまざまで、CPUやメモリーなどの性能も差が大きい。Javaのようなハードウェアやを問わない言語がなくては、こうした機器をネットワークにつないで活用することはできない。

1999年に情報家電や携帯機器向けのJavaの枠組みJ2ME（Java 2 Micro Edition）が発表されたことで、この分野でのJavaはいよいよ実用段階を迎える。その象徴となるのが、この12月に始まるiモードへのJavaの搭載だ。日本ではパソコンを超えるユーザー人口を持つ携帯電話に載ることで、Javaは一気に普及することになる。ゲームソフトなどのコンテンツをダウンロードして携帯電話で楽しめるだけでなく、Javaのセキュリティー機能を活かしたショッピングやオンライントレードなどのサービスがいつでもどこでも利用できるようになる。

求む！ Javaプログラマー

3つのカテゴリーに分けられて方向性がはっきりとした昨年から今年にかけて、Javaはインターネット時代のアプリケーション開発の本命として認識されるようになった。特にサーバー上でのJavaプログラムに対する需要は急増している。しかし、残念ながら日本ではいまいち盛り上がりに欠けているようだ。インターネットでのビジネスを考えているなら、今後Javaは無視できない。遅れをとった日本で先駆けとなってJavaの開発を始めれば、大きなチャンスをつかめるだろう。

これまでのJavaの歩み

黎明期	1995年	5月	サン・マイクロシステムズ、JavaおよびHotJavaを発表。
		9月	ネットスケープ、Javaを搭載したネットスケープナビゲーター2.0発表。
		12月	マイクロソフト、インターネット戦略発表、Javaをライセンス。
流行期	1996年	2月	JDK 1.0正式公開。
		5月	サン、オラクルなど5社がNCで提携。
		5月	JavaOS発表。
		8月	マイクロソフト、Javaを搭載したインターネットエクスプローラ3.0公開。
		10月	JavaStation発表。
		10月	JavaBeans発表。
		10月	JavaCard発表。
混乱期	1997年	2月	JDK1.1正式公開。
		4月	PersonalJava、EmbeddedJava発表。
		6月	ネットスケープ、ネットスケープコミュニケーター4.0公開。
		9月	マイクロソフト、インターネットエクスプローラ4.0公開。
		10月	サン、マイクロソフトをライセンス違反で提訴。
		12月	Enterprise JavaBeans発表。
	1998年	7月	Jiniのデモが行われる。
		10月	Java Server Pages発表。
		11月	AOL、ネットスケープを買収、サンと提携。
		12月	Java 2（JDK1.2）正式公開。
発展期	1999年	1月	Jini正式発表。
		3月	サン、NTTドコモと提携。
		6月	J2EE、J2ME発表。
	2000年	5月	JDK1.3正式公開。
		12月	iモードにJava搭載。

XML（エックス・エム・エル／eXtensible Markup Language）：文書の構造を記述するタグ言語の1つでSGMLをリニューアルしたもの。文書自身の中にタグを埋め込む形で利用され、ユーザー自身が独自のタグを定義することができるのが特徴。

chap.1

Java2を構成する3つのE

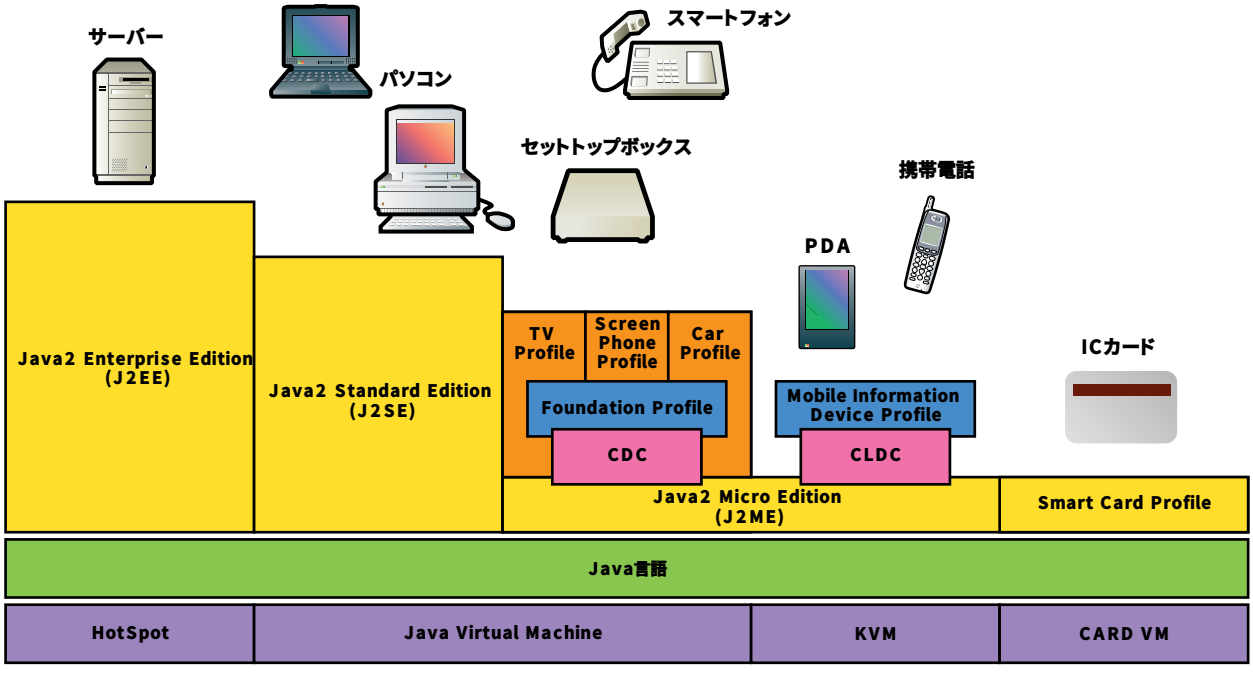
Javaが発表されてから5年。この間Javaは常に進化し続け、サーバーから携帯端末へと応用分野を拡大してきた。現在「Java」という言葉はいったい何を指すのだろうか。今のJavaがカバーしている範囲をここで確認しよう。

□ Javaによる情報機器の統一

1998年末に最新のJavaは「Java 2」と命名され、続く1999年のJavaOneではJava 2プラットフォームを3つに分類することが発表された。サーバー環境から携帯端末まで、用途も性格も違うハードウェアを同じJavaで動かしても最高のパフォーマンスが得られないので、分野や規模に応じてパフォーマンスを追求することになったのだ。これが今もJavaの基本的な枠組みとなっている。

「Java 2 Standard Edition」（J2SE）は、ワークステーションやパソコン、一部のセットトップボックスを想定しており、Java 2のコアでもある。「Java 2 Enterprise Edition」（J2EE）が想定しているのはサーバー環境で、基幹システムのような大規模な環境にも十分に機能するように綿密に調整されている。そしてセットトップボックスや携帯電話、PDAといった小さな機器のために「Java 2 Micro Edition」（J2ME）がある。

重要なのは、このページの図の一番下だ。HotSpot、JVM、KVM、CardVMのようなVM（仮想マシン）だけがハードウェアに依存する。この構造がJavaの「Write Once, Run Everyware」（一度書いたらどこでも動く）を基本的に保証しながら、幅広い情報機器でJavaが動作する仕組みを提供しているのだ。



Java 2 プラットフォーム

セットトップボックス：テレビなどに接続して各種のサービスを受けるための機器の総称。ケーブルテレビの番組を受信したり、CS放送番組を受信したり、またインターネットに接続したりするためのものなど、さまざまな種類がある。STB（Set Top Box）と書き表されることも多い。

□ Java2プラットフォーム

J2SE　Java 2 Standard Edition

Java 2の基本となるJava環境。Java 2以前に「Java」と呼ばれていたものと同じだ。JVMや基本的なAPI、拡張API、Java Plug-inなどからなる。おもにパソコンなどのクライアントコンピュータを対象とし、Javaアプレットや Javaアプリケーションが動作する環境となる。また、Java 2Dや Advanced Imaging、Java 3DなどのAPIを使えば、高度なグラフィックス操作も可能で、マルチメディア環境を実現したいなら Java Media Frameworkによって静止画だけでなく動画も扱うことができる。J2SEをサポートしたWWWブラウザーはまだないが、現在開発中のネットスケープ6にはJ2SEが含まれている。

J2EE　Java 2 Enterprise Edition

J2SEが持つクライアントサイドの機能にサーブレット、JSP（Java Server Pages）、EJB（Enterprise JavaBeans）などのサーバサイドの機能を付加し、エンタープライズコンピューティングのための環境を提供するもの（212ページ参照）。サーバーは多くのクライアントからの指示を受けてその結果を返すため、CPUとメモリーに比較的長い時間負荷がかかる。クライアントでは、処理の大半がGUI関連、つまり画面表示に使われるので、一時的な負担は大きいもののサーバーに比べてCPUやメモリーへの負荷がかかる時間は短くなる。J2EEはこうしたサーバーの特性に合わせて最適化されている。

J2ME　Java 2 Micro Edition

セットトップボックスから携帯電話までごく小さい機器をサポートし、ネットワークにつながった情報家電でJavaを動作させるための環境。ほかのJava 2プラットフォームとは違い、限られたリソースを最大限に利用してJavaを動作させる仕組みを提供する。APIはそれぞれの機器に最適化されたものが用意され、GUIやネットワークサービスの部分で多少の非互換があるものの、ほかのJava 2プラットフォームとの互換性は保証されている。携帯電話やPDAなどのためには、KVM（K Virtual Machine、Kはキロバイトの意味）と呼ばれるVMが用意され、極めて限られたリソースの中にもVMを納められる（209ページ参照）。

Java用語集

API（Application Programming Interface）
Javaプログラムを開発する際に利用できる機能のセットで、JRE（下記参照）により提供される。入出力やネットワークなどの基本的なAPIのほかに、セキュリティーやSwing（下記参照）などの拡張APIがある。

HotSpot
Java Virtual Machineを高速化させるための技術。プログラムの中で実行速度においてボトルネックになっているところを見つけて最適化する。J2EEに含まれていたが、J2SE 1.3でもサポートされた。

JavaBeans
Java環境に用意されたコンポーネント技術。JavaBeansによって、どんなプラットフォームでも再利用可能なソフトウェアの部品を開発できる。GUIを活用した開発ツールで利用することが想定されている。

Java Plug-in
WWWブラウザーに含まれるJava環境を使わずに、サン・マイクロシステムズの最新のJavaでアプレットを動作させるための仕組み。ActiveXコントロールやネットスケーププラグインとして動作する。

JCP（Java Community Process）
オープンな場で業界の支持を得ながらJavaの仕様を制定するためにサン・マイクロシステムズが用意した手続き。業界の各社が参加して、Javaの仕様に意見を反映させることができる。

JDK（Java Development Kit）
Javaプログラム開発のための基本キット。コンパイラーとJREからなる。以前はJava環境のことも指していたが、現在JDKは「Java 2 SDK」、Java環境全体は「Java 2プラットフォーム」と呼ばれる。

JRE（Java Runtime Environment）
JDKで開発されたJavaソフトウェアを実行するランタイム（実行環境）。Core API、拡張API、JVM、HotSpot、Java Plig-inで構成されており、Javaソフトウェアを実行する環境のみを提供する。

Profile
特定の機器のための特化されたAPIセット。情報機器にはさまざまな種類のものがあるが、テレビや携帯電話などで共通する機能を定義することで、種類が同じ機器なら同じAPIが使えるようになる。

Swing
Javaプログラムから利用できる高機能なグラフィカルユーザーインターフェイス用のAPI。異なるプラットフォームで共通のインターフェイスを持ったアプリケーションが作成できる。

VM（Virtual Machine）
Javaの実行環境のうち、「仮想マシン」と呼ばれる部分。Javaコードを解釈しながら各OS用のコードに変換して実行する。どんなプラットフォームでも、VMがあればJavaプログラムが実行できる。

GUI（ジー・ユー・アイ／Graphical User Interface）：ユーザーに対する情報の表示にグラフィックを多用して、多くの基本的な操作をマウスなどのポインティングデバイスによって行えるインターフェイスのこと。GUIと略して呼ばれることが多い。

Javaがケータイを動かす

chap.2

いよいよ12月からJavaが搭載されたiモード端末が登場する。携帯電話に代表される、インターネットにつながった情報機器は、今後もっともJavaの活躍が期待できる分野だ。いよいよ実用段階に入った「ケータイJava」に注目しよう。

韓国LG TeleComのJavaを搭載した携帯電話 I-Folder

モトローラのiDEN i3000 plus

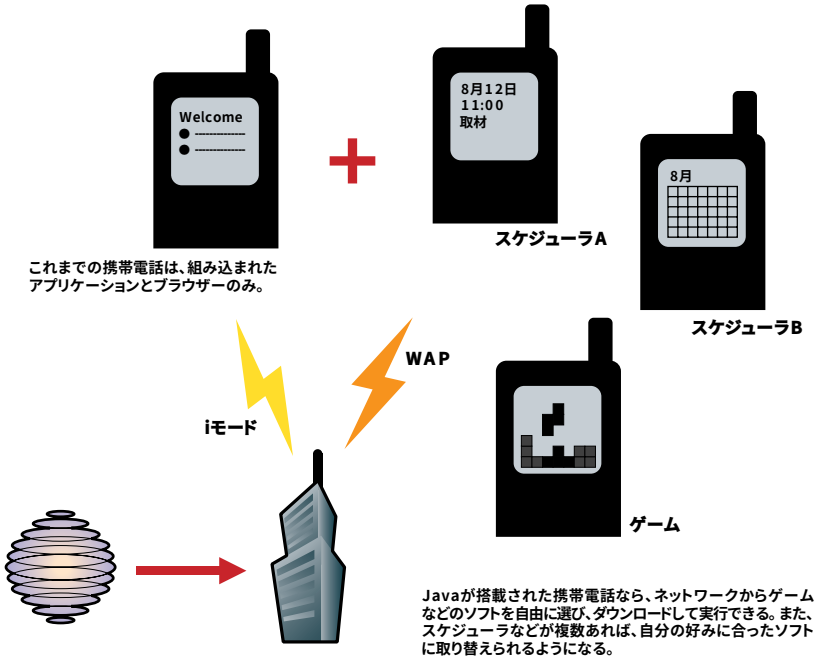
BLACKBERRYのRIMシリーズ

□ Javaが情報機器を変える

Javaが小型機器の制御用言語から出発したこともあり、モバイル機器に対するJavaのサポートはかなり力の入ったものになっている。また、サン・マイクロシステムズ社をはじめJCP（Java Community Process）に参加している各社は、この分野の重要性を認識している。

なぜこれほどまでに携帯端末が注目されるのだろうか。それは、パソコンに限られていたインターネット利用がiモードのような携帯電話によって爆発的に増えているからだ。インターネットと相性がよく、どんな機器でも動作するJavaがあれば、インターネット上のサービスを利用したソフトウェアの開発が簡単になり、だれもがそれをインターネットを通じて入手できるようになる。Javaのセキュリティー機能によって、携帯電話によるeコマースへの参加も増えるだろう。

携帯電話に限らず、ゲーム機や家電に関しても同じだ。特定のプログラムしか動かせなかった機器がインターネットとJavaによって無限の可能性を持つようになるのだ。



携帯電話＋Javaの可能性

eコマース（イーコマース）：電子商取引のこと。ネットワーク上で電子的に決済情報を交換して行う商業上の売買行為のこと。小売市場に限らず、企業対企業の取引や個人対個人の取引も含む。

□ J2MEとCLDC

Java 2プラットフォームでは、セットトップボックスやPDA、携帯電話といったリソースの限られた小型の機器向けにJ2ME（Java 2 Micro Edition）が用意されている。J2MEがどんな構成を取っているのかを見てみよう。

携帯電話とセットトップボックスでは、同じようにJavaを動作させることはできない。そこで、JCP（Java Community Process）によってCDC（The Connected, Device Configuration）とCLDC（The Connected, Limited Device Configuration）という、J2MEを特定の機器に特化させた仕様が定義されている（206ページの図を参照）。CDCはセットトップボックス、スマートフォン、カーナビのような固定して使われる情報機器を、CLDCはPDAや携帯電話、双方向ページャーのような小さなモバイル端末を対象としている。

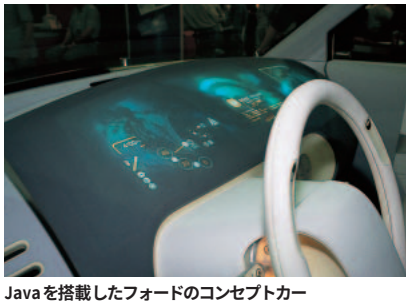
CLDCで想定される機器のリソースは、128KBから512KBのメモリーと周波数が16MHzから32MHzのCPUとなっており、極小モバイル機器をターゲットとしていることがわかる。CLDCはJava言語とVM、コアJavaライブラリー、入出力、ネットワーク、セキュリティー、国際化のみをサポートし、ユーザーインターフェイスなどについては定義しない。ネットワークにつながった機器が対象だが、携帯電話は直接インターネットに接続するわけではないので、TCP/IPプロトコルに基づいていなくてもいい。

また、CLDCはVM（仮想マシン）にKVMという小型機器向けのものが用意されていることが特徴だ。KVMの「K」は「キロバイト」の意味で、これは、数10KBの単位で計られるような大きさに収まるものであることを意味している。

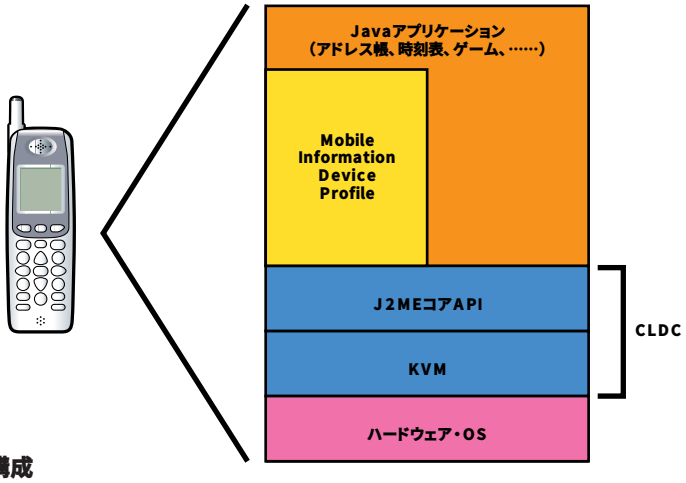
CDCやCLDCの上には、セットトップボックスや携帯端末といった機器のタイプごとにプロファイル（Profile）というAPIセットが提供される。J2MEとCLDCが対象にしている携帯端末向けのプロファイルとして、MIDP（Mobile Information Device Profile）がある。

このようにJ2MEは何重もの構成を取ることで、変化の激しく範囲の広い機器を柔軟にサポートしながら、基本的な部分ではJavaの互換性が保たれるようになっている。


シャープの液晶マルチメディアホン


Javaを搭載したフォードのコンセプトカー

ドリームキャストとPlanetweb製のJavaブラウザー



J2MEの構成

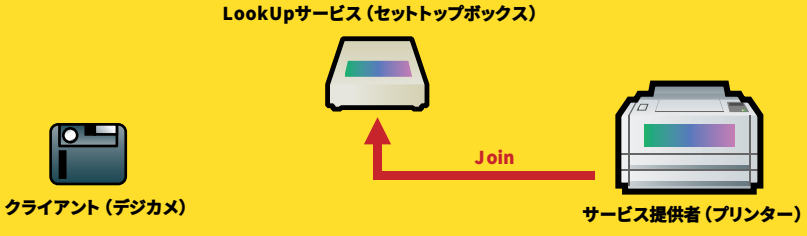
TCP/IP（ティー・シー・ピー・アイ・ピー／Transmission Control Protocol/Internet Protocol）：インターネットでデータをやり取りするために標準的に用いられるプロトコル（通信規約）。HTTPやFTPなどの基盤となるプロトコルである。

□あらゆる機器を協調させるJini

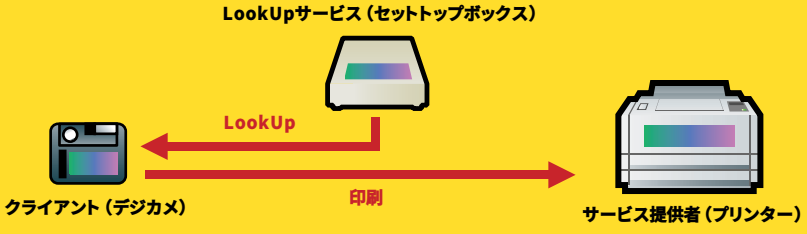
Jiniの仕組み

❶サービス提供者はLookUpサービスを探す。

❷サービス提供者はLookUpサービスに自分の情報を登録する。



❸LookUpサービスから送られた情報によって、クライアントとサービス提供者が直接対話する。



まず、「Discovery」によってネットワークにJiniデバイスが存在するのか（つまりこのネットワークはJiniネットワークか）を探り、LookUpサービスを見つけ出す（図上）。次に「Join」によってLookUpサービスに登録し、その連合体の一部になる（図中）。最後にLookUpサービスが、必要なサービスを探し出し提供するという一連の流れで動作する（図下）。

J2MEによって家電がJavaを搭載する環境が整ったが、冷蔵庫のような家電を買ってきて、IPアドレスなどのインターネットの設定を行うことは非現実的だ。情報機器が身近になればなるほど、わずらわしい作業を減らして、「つなぐだけでいい」という使い勝手にすることが重要になる。

この問題に関してもJavaは確実な解答を持っている。それがJiniだ。Jiniは、ネットワークにつながるあらゆる機器を「federation」（連合体）とするための仕組みを提供する。Jiniでは、ネットワークにつながれたあらゆる機器（Jiniデバイス）がサーバーとなり、場所を問わずにJavaのサービスを受けたり提供したりできる。

Jiniの基礎は、ネットワーク上で自由に移動できるオブジェクトを交換し、そのオブジェクトを使ってサービスを提供する「分散オブジェクト」だ。サービスとオブジェクトの出会いと言える制御を行うものはLookUpサービスと呼ばれる（左図参照）。

Jiniを利用すれば、機器がどこにあるのかを気にせずにさまざまなサービスを受けることができる。たとえばストレージサービスでは、Jiniに対応したハードディスクをネットワークに接続するだけで一切の設定なしにファイルサービスが提供できる。また、Jiniに対応したビデオがあれば世界中のどこからでも録画の予約が可能になる。

Jiniによってバーコード読み取り機と連係し、食品を管理するWhirlpool製の冷蔵庫

Jiniで連携するデジタルカメラとPDA、プリンター

IPアドレス：インターネットに接続されたコンピュータを識別するために付けられる、基本的に世界でよっしかない番号。IPアドレスにわかりやすい名前を付けたのがホスト名。

ジョン・ゲイジ

iモードを語る

7月19日、サン・マイクロシステムズのチーフリサーチャー、ジョン・ゲイジ氏による記者団との会見が行われた。九州沖縄サミットに合わせての来日で、シスコ・システムズのジョン・チェンバースCEOやソニーの出井伸之会長らとともに首相官邸での「IT革命」についての意見交換を行ったゲイジ氏は、iモードを手にして登場した。

Photo:Watari Tokuhiro

昨日このiモードを渡されました。サミットにやってきた総理大臣、大統領、報道関係者はこの素晴らしい携帯電話を手にします。このiモードの最初のページには、8か国の国旗が表示されます。これを見て、日本が世界をリードしていることを実感するでしょう。

この12月にはiモードにJavaが搭載されると聞いています。昨晩ホテルでヤフーに登録してある自分のカレンダーをiモードで見ることができました。また、ヤフーのほとんどのページを見ることもできました。iモード用に1万7千ものウェブページが作られているのは、コンパクトHTMLという標準を採用しているからです。そのおかげで、だれでも新しいコンテンツを作って届けることができるのです。Javaが搭載されれば、JavaOneにやってきた100か国の300の企業が自分たちのプログラムをiモードで動かすことができるようになります。

Javaはセットトップボックスや携帯電話に次々と採用されています。また、ソニーのプレーステイションにも搭載されるでしょう。Javaの力があれば、こうした異なる機器の上で動くコンテンツを開発できるのです。今の携帯電話には限られた機能しかありませんが、iモードではさまざまな新しいプログラムを取り替えられるようになります。現在、ホテルでは名前とクレジットカードの番号を伝えますが、これからは名前と携帯電話の番号を伝えるようになるでしょう。携帯電話で支払いを行うようになるからです。

JavaOneでも話した私の大好きなアイデアをお話ししましょう。私はソニーのMDビデオカメラを持っていますが、カメラの中でJavaが動けば、カメラをWWWサーバーにするプログラムを書くことができます。ビデオを撮影し、インターネットに流してブラウザーで閲覧できるようになります。iモードを使ってサンフランシスコにあるビデオのIPアドレスを指定すれば、ここにいながらiモードの画面でビデオを鑑賞できるようになります。このようにして、我々は突然、ばらばらだったテレビ、電話、コンピューターなどの開発者が一緒になってお互いの産業に何か新しいものを付け加える様子を見ることになります。

Jiniについてお話ししましょう。Jiniは、何かの機能を持ったJavaプログラムがネットワーク越しにサービスを提供し、無数の独立した機器がコミュニケーションを行うというものです。Jiniデバイスは、ピアツーピア（一対一）で会話をします。すべての機器が平等になるわけです。Jiniを実現するチップは値段の高いものではありませんから、あらゆる場所の機器を賢くすることができます。

現在は、電気製品を買ってくるとつなげる場所を探して回らなければなりません。IEEE1394を使えば、機器をどこかにつなぐだけで何百メガの速度でネットワークに接続できます。さらに最新のワイアレス技術を使えば、家に機器を持ってくるだけで、その機器はほかの機器と友達になることができます。Jiniでは自分ができることをその機器自身が知っています。たとえば、Jini対応の携帯電話を持って車に載ると、携帯電話が車に「私は電話です」と語りかけます。すると車は「具合が悪い。オイルが必要だ。電話をかけてくれ」と話します。携帯電話は「もちろん」と答えるという具合です。冷蔵庫や電子レンジなどあらゆる情報機器でも同じようなことができます。

IEEE1394（アイ・トリプル・イーイチサンキューヨン／Institute of Electrical and Electronics Engineers 1394）：シリアルインターフェイス規格の1つ。最大で63台の機器を接続でき、転送速度は現在最大で400Mbps。機器の動作中でも抜き差しできる。

chap.3

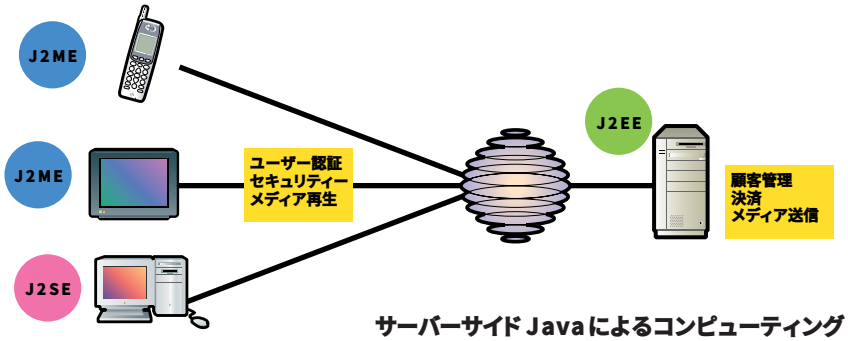
Javaが eコマースを支える

WWWブラウザーのユーザーがアプレットに飽きてしまったころ、Javaはサーバーサイドに活躍の場を見出した。今ではサーバー上のアプリケーションを開発するのにJavaを使うのは常識だ。eコマースに欠かせないJavaの技術を確かめよう。

□サーバーサイドJavaの力

Java 2プラットフォームにおいて、サーバー側のJava技術を統合したものがJ2EE（Java 2 Enterprise Edition）だ。J2EEが存在する意義は、真の分散型ネットワークのプラットフォームが整備された点にある。J2EEは、ただ企業内の情報システムにJavaを浸透させようとしているわけではなく、グローバルな情報システムを強力に発展させる力を秘めている。

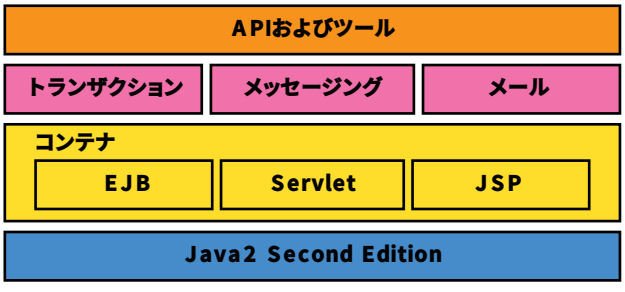
旧来のメインフレームとダム端末のような関係とは違い、Javaによるコンピューティングでは、単純にサーバーの実行結果を端末に表示するだけではない。クライアント側にJavaがあれば、ユーザーの認証やプログラムの安全性確認などのセキュリティー処理、マルチメディアの再生などさまざまな処理ができる。J2EEによって、単純なB2Cにとどまらず、ブロードバンドを利用したビデオオンデマンドなどのマルチメディアを含めた、トータルエンタープライズコンピューティングとも言うべきスタイルが今まさに始まろうとしている。



サーバーサイドJavaによるコンピューティング

□J2EEの構成

J2EEは、J2SEのクライアントサイドの機能にサーバーサイドの技術を付加して、エンタープライズコンピューティングに必要な機能を提供している。J2EEには、サーブレット（Servlet）、JSP（Java Server Pages）、EJB（Enterprise Java Beans）を動作させる「コンテナ」というサーバーサイドのJava実行環境が搭載され、トランザクションやメッセージング、メールなどのAPIを用途に応じて利用するという形を取る。



J2EEの構成

通信インフラへの接続としては、TCP/IP、HTTP、HTML、SSL（ウェブ通信の暗号化）、IIOP（CORBAアプリケーションの通信プロトコル）が備わり、エンタープライズコンピューティングに不可欠なさまざまなシステムとの連携を支える。

このようなJ2EEの機能を組み合わせれば、複雑なプログラミングを行わなくてもサーバー上のアプリケーションが容易に開発できるようになる。「Write Once, Run Anywhere」という言葉に示されるJavaの互換性も保証されるので、J2EEをサポートしたサーバー同士なら開発したアプリケーションを再利用できるようになる。

メインフレーム：おもに企業や大学などで用いられてきた汎用大型コンピュータのこと。

□ J2EEのキーテクノロジー

J2EEとして統合されたサーバーサイドJavaのキーとなる技術が、サーブレット（Servlet）、JSP（Java Server Pages）、EJB（Enterprise Java Beans）だ。この3つは、サーバーサイドコンピューティングを進めるうえで重要な役割を果たす。

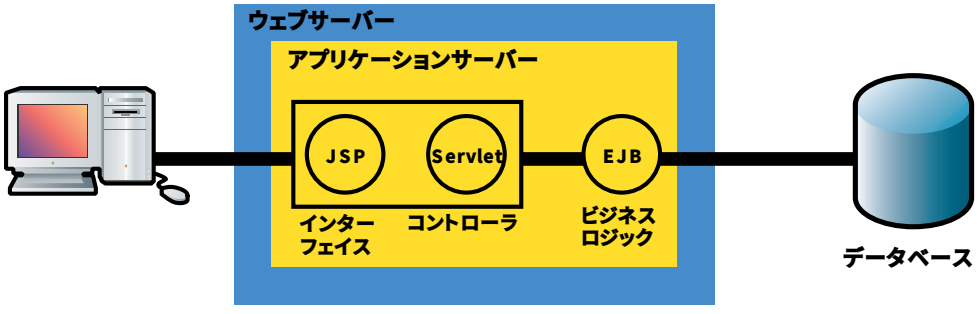
サーブレットは、これまでのCGIに置き換わるJavaのプログラムで、言わばサーバー上で動作するアプレットだ。HTTPを通してリクエストを受けると起動し、結果を動的に作成してウェブページとしてクライアントに返す。JSPは、テキストで書かれたファイルで、HTMLの中にJavaの命令をスクリプトとして埋め込んでおくと、サーバー上でその命令を実行してHTMLとともに結果をクライアントに返す。

サーブレットとJSPを役割分担をさせながら組み合わせて使うのが、サーバーサイドJavaの主流になっている。サーブレットだけでウェブページを生成すると、HTMLのタグをプログラムの中に延々と埋め込むことになってしまい、ウェブ作成の効率が悪くなる。HTMLのデザインに変更が生じた場合にサーブレット自体を修正してコンパイルしなおすことになるからだ。また、デザインを担うJSPの中に長いプログラムを書き込むのも効率が悪い。HTMLをなるべくJSPに置いて、サーブレットから見た目を切り離せば、プログラムを変更せずにデザインだけを更新していくことができる。JSPの機能は地味ではあるが、ロジックに手を入れるリスクを減少させるため、システムの規模が大きければ大きいほど切り離しの効果が出る。

サーブレットからHTMLの記述を取り除き、JSPに機能分担させることと同じことがサーブレットとEJBの間で成り立つ。それは、EJBを使ってビジネスロジックを独立させるという方法だ。EJBの詳細は後述するが、ビジネスロジックとは、データベースと結び付いた、在庫管理、顧客管理、販売注文、請求書作成などのプログラムを指す。ビジネスロジックとビジネスロジックを効果的に動作させるためのプログラムは独立させる必要がある。ビジネスロジックの変更のためにシステムの基本設計が修正されることが避けられるからだ。JSPによって見た目を独立させ、EJBによってビジネスロジックを独立させることにより、サーブレットはこれらのコントロールを行う役目に専念できる。ビジネスロジックが変わっても、見た目が変わっても、システムは変わらないという環境が実現でき、サーバーサイドの実行環境が極めてシンプルになる。

これは、単純に効率だけの問題ではなく、今日のあまりにも速いビジネスのスピードに情報システムを追従させるための最良の方法なのだ。

JSP、サーブレット、EJBの組み合わせ



インターフェイス:デザイナーはプログラムを意識せずにデザインに専念する。
コントローラ:プログラマーはビジネスロジックを組み合わせてプログラミングを行う。
ビジネスロジック:プログラマーはビジネスロジックを部品化して再利用できるようにする。

おもなアプリケーションサーバー

製品名	発売元	価格	Jump
BEA WebLogic Server	日本BEAシステムズ（株）	1,980,000円〜	www.beasys.co.jp
Inprise Application Server 4	インプライズ（株）	1,500,000円〜	www.inprise.co.jp
JRun Server 3.0	（株）シリウス	Professional版198,000円〜、Enterprise版1,200,000円〜	www.sirius.co.jp
iPlanet Netscape Application Server 4.0	iPlanet E-Commerce Solutions	5,103,000円〜	ja.iplanet.com
Oracle Application Server R4.0.8	日本オラクル（株）	800,000円、Enterprise版2,500,000円	www.oracle.co.jp
SilverStream Application Server 3.0	ニチメンデータシステム（株）	Enterprise版 2,300,000円〜	www.nichimen-nds.co.jp
Tomcat※	The Jakarta Project	フリーソフトウェア	jakarta.apache.org
WebSphere Application Server　V3.5	日本アイ・ビー・エム（株）	スタンダード版127,200円〜、アドバンスド版1,200,000円〜	www.jp.ibm.com

※ApacheのServlet/JSP用アドオン。The Jakarta ProjectはオープンソースのサーバーサイドJavaの開発グループ。

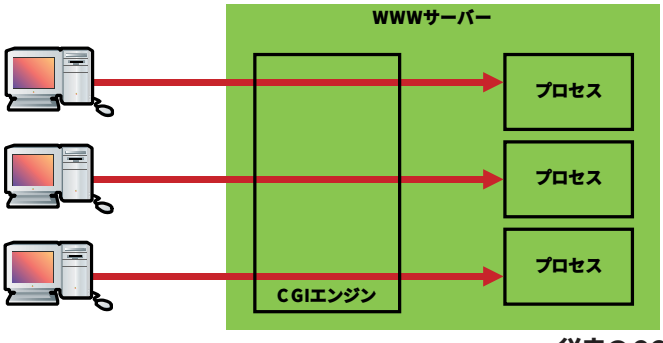
CGI（シー・ジー・アイ／Common Gateway Interface）：WWWサーバーで、外部プログラムを起動して、その結果をWWWクライアント（ブラウザー）に返す仕組み。ユーザーのリクエストによってウェブページの表示内容を変えたり、ユーザーの入力したフォームのデータをサーバー側へ読み込んだりできる。

□サーブレットの利点

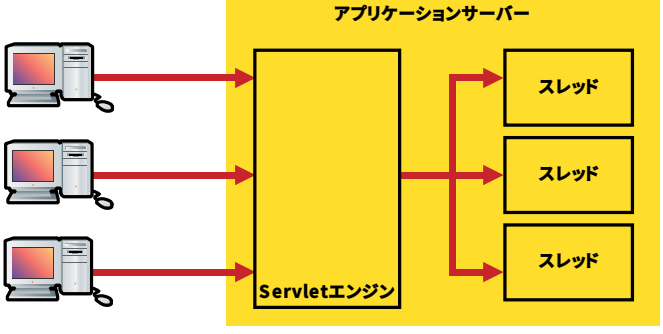
サーバーサイドJavaの代表的な技術がJSPとサーブレットだ。サーブレットは、サーバー上のアプレットと言える。アプレットがクライアントにダウンロードされて実行されるのに対し、サーブレットはサーバー上で実行され結果をクライアントに戻している。

従来のウェブでは、CGIがサーバー上のアプリケーションを実行していたが、CGIは大規模かつミッションクリティカルな要求にパフォーマンスと信頼性の両面で答えられるものではなかった。この問題は、CGIの構造に原因がある。CGIは、1つのリクエストを1つのプロセスで処理することと、セッションの維持が難しいことが最大の弱点になっている。プロセスの数が増えるとともに、システム全体のパフォーマンスが低下する。

サーブレットは、Javaのネットワーク言語としての適性とプログラム再利用という手法によって、パフォーマンスと信頼性の問題を解決している。サーブレットはCGIと異なり、1つのリクエストを1つのスレッドで処理している。プロセスよりも小さな処理単位であるスレッドで処理することで、マルチタスキングの効率を上げ、システム全体のパフォーマンスを維持する結果につながる。



従来のCG



サーブレットによるパフォーマンスの向上

□HTMLの記述を受け持つJSP

JSPはテキスト形式のファイルで、ふつうのHTMLの中にJSP固有の言わば拡張タグを埋め込むことができる。この拡張タグによってEJBやJavaBeansなどのコンポーネントを呼び出したり、サーブレットからの要求を受け付けたりする。サーブレットと役割分担をすることで、不必要なコンパイルを減らし、パフォーマンスの向上とウェブページの表現力や生産性の向上を実現している。左がJSPの簡単な例で、WWWブラウザーの種類を調べて表示を変更している。「<%」から「%>」で囲われた部分がJSPのタグで、ちょっとしたJavaプログラムを直接記述したり、条件に応じて別のファイルをページの中に取り込んだりできる。

```
<%@ page language=="java" info="Example JSP #1" %>
<HTML>
<BODY>
<%! String agent; %>
<%
agent = request.getHeader("User-Agent");
if ( agent.startsWith("Mozilla/4.0") {
%>
<%-- Return content for 4.0 browsers --%>
<%@ include file="ver4.html" %>
<%
} else {
%>
<%-- Return content for other/unknown browsers --%>
<%@ include file="other.html" %>
<%
}
%>
</BODY>
</HTML>
```

JSPの例

プロセス：プログラムの実行単位。1つのプログラムを起動するたびに、1つのプロセスがメモリ上に作成される。UNIXやウィンドウズなどのOSでは、複数のプロセスが独立しながら平行して動作する。

□サーブレットでCGIを置き換える

それではここで、サーブレットのサンプルを見てみよう。HTMLのフォームから送られたデータを取り出して、そのまま表示するという簡単なものだ。

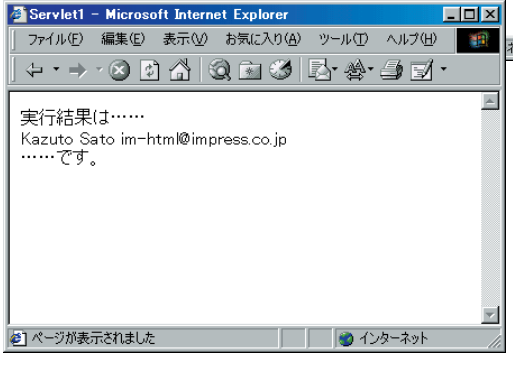
まず、Javaソースの先頭でこのアプリケーションのパッケージ名を宣言し❶、サーブレットを使うために必要なパッケージをインポートする❷。アプレットではAppletクラスのサブクラスを作るように、サーブレットではHttpServletクラスのサブクラス「Servlet1」を作成する❸。初期設定用のメソッド「init」❹と、フォームからデータが送られたときに呼び出されるメソッド「doPost」❺を記述する。「doPost」では、フォームのデータを取り出して変数に入れ❻、そのままHTMLのタグとともに出力している❼。だいたいの雰囲気はつかめただろうか。

なお、JSPやサーブレットを動作させるにはサーブレットエンジンが必要となるが、Java 2の各プラットフォームには装備されておらず、TomcatやJRunなどのサーブレットエンジンか、BEA WebLogicやOracle Application Serverなどのアプリケーションサーバー（213ページの表参照）を使う必要がある。

```
package sample.servlet;                    ❶

import java.io.*;                          ❷
import javax.servlet.*;
import javax.servlet.http.*;

// サーブレットクラス（HttpServlet.class）のサブクラスとして宣言
public class Servlet1 extends HttpServlet {  ❸

 // 初期設定
 public void init(ServletConfig config) throws ServletException {  ❹
   super.init(config);
 }

 // Submitボタンが押された（Postされた）ときに呼び出されるメソッド
 public void doPost(HttpServletRequest request,
    HttpServletResponse response)                                  ❺
    throws ServletException, IOException {
  // 変数の初期設定
  String var0 = "";
  String var1 = "";
  // HTMLページから送られたパラメータを変数に格納
  try {
    var0 = request.getParameter("name");                           ❻
    var1 = request.getParameter("mail");
  }
  // 例外処理
  catch(Exception e) {
    e.printStackTrace();
  }
  // 出力するページのタイプを設定
  response.setContentType("text/html");
  PrintWriter out = new PrintWriter (response.getOutputStream());
  // HTMLの出力
  out.println("<HTML>");                                          ❼
  out.println("<HEAD><TITLE>Servlet1</TITLE></HEAD>");
  out.println("<BODY>");
  out.println("実行結果は……<BR>");

  // 入力された文字列を結合して表示
  out.println(var0 + " " + var1 + "<BR>");
  out.println("……です。<BR>");
  out.println("</BODY></HTML>");
  out.close();
 }
}
```

サーブレットの例

フォームを使ってウェブページからサーブレットを呼び出す。



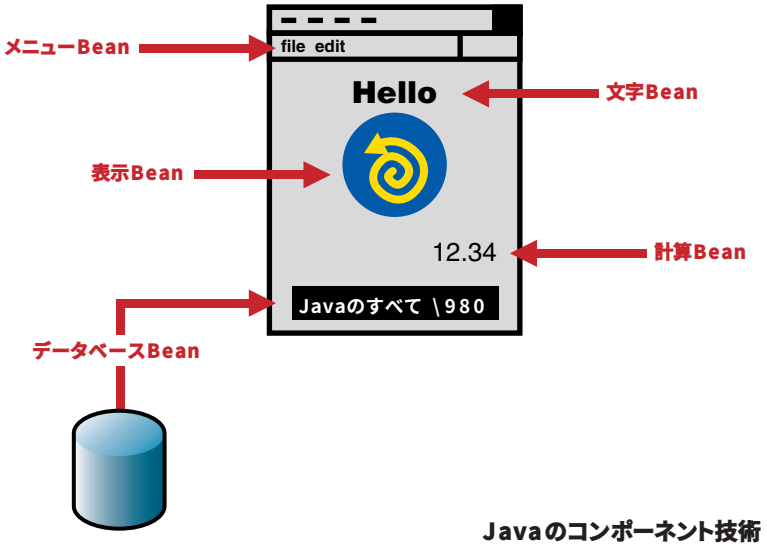
サーブレットによって返された実行結果。

```
<HTML>
<HEAD><TITLE>Servlet1</TITLE></HEAD>
<BODY>

<FORM ACTION=http://www.****.co.jp/sample.servlet.Servlet1 METHOD="POST">
お名前： <INPUT TYPE="text" NAME="name">
メールアドレス： <INPUT TYPE="text" NAME="mail">
<INPUT TYPE="submit" VALUE="送信">
</FORM>

</BODY>
</HTML>
```

サーブレットを呼び出すHTML

クラス：オブジェクト指向のプログラミング言語において、オブジェクトの機能やデータを定義したもの。クラスの内容を隠蔽して保守性を高めたり、新しいクラスを派生させたりできる。

□コンポーネント技術EJB



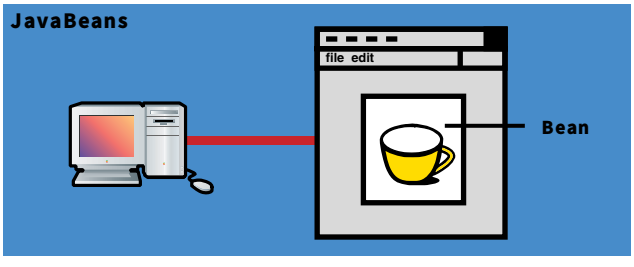
Javaのコンポーネント技術

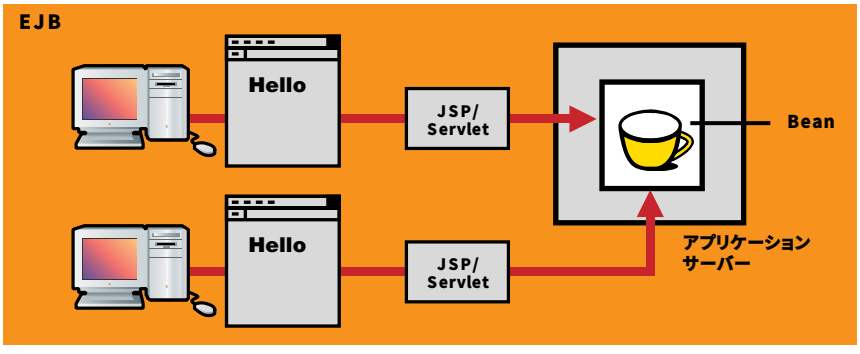
EJB（Enterprise JavaBeans）は、JavaBeansのコンポーネント技術をサーバサイドJavaに応用したもので、エンタープライズコンピューティングに不可欠な機能だ。

コンポーネント技術とは、プログラムをある程度のサイズや機能に集約したソフトウェアの部品として利用するものだ。コンポーネント技術を利用することで、Javaのメリットの1つであるプログラムの再利用が可能になり、効率が向上する。

たとえば、ふつうのプログラムでは、ファイルにアクセスしたい場合には入出力関数の呼び出しなどを記述する必要がある。しかしファイル機能をコンポーネント化すると、そのファイルアクセス用のBeanを使えば自分でプログラムを書かなくて済む。ブロックのように組み合わせていくだけでプログラムが完成するのだ。

□JavaBeansとEJBの違い





JavaBeansとEJB

EJBとこれまでのJavaBeansとの違いは、大ざっぱに言えばアプレットとサーブレットのように、クライアントサイドかサーバーサイドかということだ。ただし、コンポーネントという概念を除いてEJBはJavaBeansとまったく違うものになっている。EJBは複数のアプリケーション間で共有が可能であり、カスタマイズの手法も違う。

JavaBeansはBeaninfoクラスやプロパティーエディターを使用するが、EJBは「Deployment Descriptor」という環境設定ファイルでカスタマイズを行う。これによってBean自体のデータに変更を加えることなくBeanの属性を変えることができる。このDeployment Descriptorには、Beanのクラス名、トランザクション、セキュリティー、ホームとリモート両方のインターフェイスに関する情報を記述する。このDeployment DescriptorによってEJBのBeanの性格が決まる。

トランザクション：「処理」の意味。コンピューターではクライアントからサーバーへの要求によって発生するデータベース処理を指すことが多い。

□ EJBによるアプリケーション開発

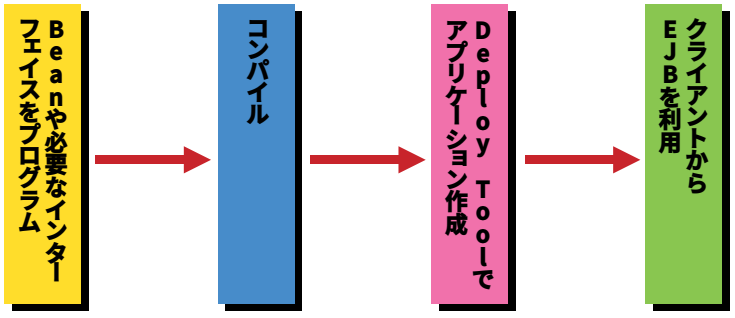
ここでEJBに関してもう少し詳しく見てみよう。EJBにはSession BeanとEntity Beanの2種類のBeanがある。Session Beanはセッション管理（たとえば、買い物や預金引き出しの手続き）を行い、Entity Beanはデータ（たとえば、商品の在庫や預金）を永続的に保持するBeanだ。Session BeanにはStatelessとStatefullの2種類が、Entity BeanにはBMPとCMPの2種類がある（下記表参照）。

Beanはアプリケーションサーバーの EJBコンテナの中で管理される。このため、データベースと接続する方法やセキュリティーなどめんどうな部分をEJBコンテナの設定にまかせてしまえるので、開発者はビジネスロジックの構築に集中できる。また、作成されたBeanは、EJBコンテナがJ2EEの仕様に従っているなら、どんなプラットフォームの上でも再利用できるものになる。

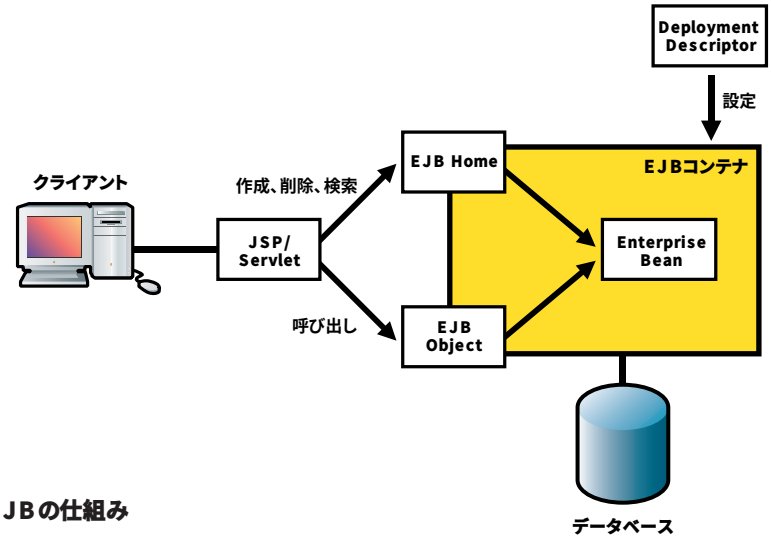
実際にEJBコンポーネント（Bean）を作成する手順を見てみよう。作成の際にはBeanそのものだけでなく、EJB Home、EJB Objectというインターフェイスを作成しなければならない。EJBを利用するサーブレットなどのプログラムは、この2つのインターフェイスを通じて間接的にBeanを利用することになる。これによって、BeanがEJBコンテナの中でどのように管理されているかを知る必要がなくなり、ビジネスロジックを分離して利用するEJBの利点が高まることになる。

2つのインターフェイスのうち、EJB ObjectはBeanのメソッドを呼び出すためのインターフェイスを提供し、EJB HomeはEJB Objectを作成するときなどに必要なインターフェイスを提供する。

Bean本体とEJB Home、EJB Objectそれぞれのソースをコンパイルして3つのclassファイルを作成したら、J2EEに含まれる「Deployment Tool」を使用してEJBアプリケーションを作成し、deploy（EJBコンテナの中に配備）を開始する。これで作成したEJBコンポーネントが利用できるようになる。



EJB開発の手順



EJBの仕組み

EJBの種類

Session Beans	サービスを提供するだけで、データを永続化しない。預金の引き出し処理など。	Stateless	内部状態を保持する。
		Stateful	内部状態を保持しない。
Entity Beans	システムがシャットダウンしても、データは継続する。銀行口座の残金など。	BMP (Bean-Managed Persistence)	Beanの状態は、Bean自身でデータベースに保存。
		CMP（Container-Managed Persistence）	Beanの状態は、EJBコンテナがデータベースに保存。

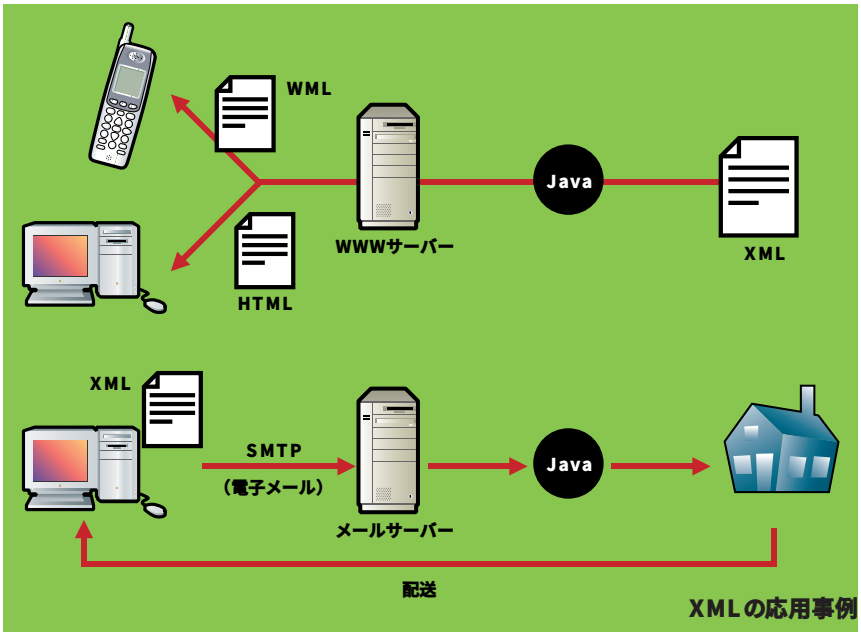
コンパイル：プログラムのソースファイルを実行形式のファイルに変換すること。

chap.4

XML＋JavaがB2Bを加速させる

B2Bなどの電子商取引では、「データの交換」が日常的な作業になる。そこで注目を集めているのが、あらゆるデータを表現できるXMLと、どこでも動くJavaの組み合わせだ。JavaとXMLの関係は今どうなっているのかを見てみよう。

□データ交換に最適なXML



XMLの応用事例

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<magazine>
  <item>
    <name>インターネットマガジン</name>
    <year>2000</year><month>10</month>
    <price>980</price>
    <description>
    特集：月々5万円から使える「データセンター完全導入ガイド」
    </description>
  </item>
  <item>
    <name>DOS/V POWER REPORT</name>
    <year>2000</year><month>10</month>
    <description>
    第1特集：最新マザーボード100選 2000年上半期
    </description>
    <price>980</price>
  </item>
</magazine>
```

XMLの例

XMLは構造化された文書だ。構造化された文書にはさまざまな使い方がある。たとえば、EJBはその性質といえるプロパティーをXMLで持っている。さまざまな設定ファイルとしてもXMLは有効だ。現在では多くのデータベースがXMLをサポートしている。

XMLの最大の特徴は互換性だ。コンピュータで処理できるように標準化されており、また左下の例のように人間にも読めるテキスト形式になっている。XMLなら、プラットフォームが変わってもデータの変更が必要ないシステムが作れる。すべてのデータベースになじむ構造になっているのだ。さまざまなプラットフォームが入り乱れている環境下では、XMLでしかデータ互換を実現できない。XMLは現在のウェブコンピューティングには欠かせないものと言える。

XMLの応用例を考えてみよう。左の図を見てほしい。サーバー上にXMLでデータを蓄えておけば、必要に応じてHTMLに変換してWWWブラウザーから閲覧できるようになる。パソコンだけでなく、たとえばWAPをサポートした携帯電話からアクセスがあったら、WMLに変換すればいい。また、パソコン上でXMLを使って商品データを管理しておき、在庫が不足したら電子メールなどを使って発注する。業者は受け取ったXMLをから必要なデータを抽出して、商品を配送する。こうした例では、XMLを変換したりデータを取り出したりするのにJavaが活躍する。互換性が高くインターネットに適したXMLとJavaを使えば、B2Bにおける自動処理システムを簡単に作成できるだろう。

B2B（ビー・ツー・ビー／Business to Business）：企業間で行われる電子商取引のこと。

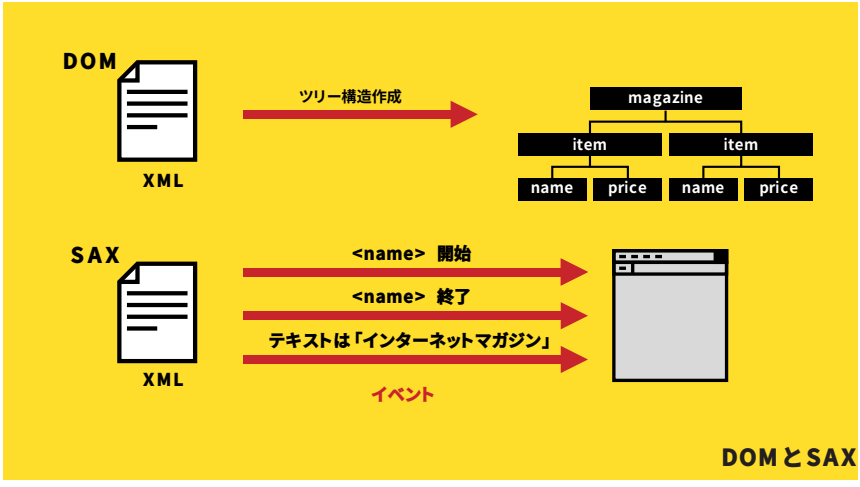
□ XMLパーサーに使われるJava

アプリケーションでXMLを扱うには、XMLパーサーというプログラムを使う必要がある。現在公開されているXMLパーサーの多くはJavaで書かれており、このことからもJavaとXMLは相性がいいことがわかる。

XMLパーサーの多くはDOMとSAXという有力なAPIをサポートする。DOM（Document Object Model）はW3Cで標準化されているAPIで、文書をツリー構造に変換する。SAX（Simple API for XML）は文書を解析しながらタグを通知するAPIで、メモリーに制限がある場合に適している。

XMLパーサーは、Apache.orgやオラクルから入手できる。オラクルは、XMLパーサーを含むXDK（Oracle XML Developer's Kit）というツールキットを提供している Jump。XDKには、XMLからJavaのクラスを作成するOracle XML Class Generatorも含まれている。

Jump www.oracle.co.jp/download/



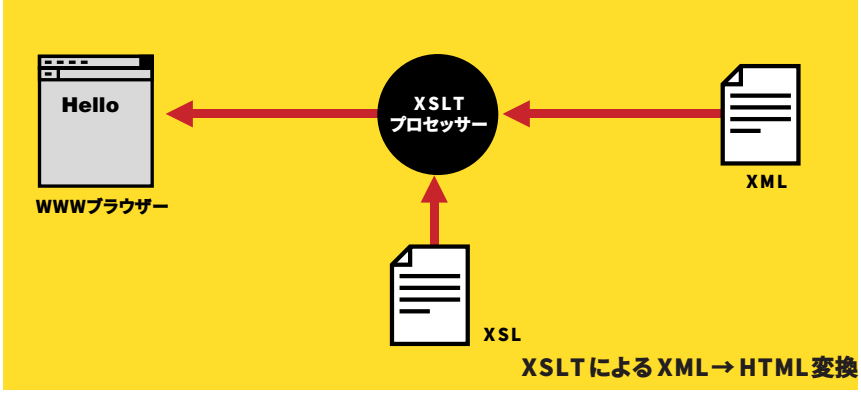
DOMとSAX

□ XSLTによるブラウザーへの表示

XML文書はそのままではWWWブラウザーで閲覧できない。XMLを「見える」ようにするには、何らかのスタイルシート言語が必要だ。XSL（Extensible Stylesheet Language）は、スタイルシート言語の1つで、XSL自体もXMLで作られている。XSLはW3Cで策定中だが、XSLの一部であるXSLT（XSL Transformations）は、すでに正式仕様が決まっている。XSLTは、XML文書をほかのXML文書やHTMLに変換するための言語だ。

テンプレートとなるXSLファイルを用意して、XSLTプロセッサーを通してXMLからHTMLを自動生成すれば、XMLをサポートしていない古いブラウザーでもデータが閲覧できるようになる。XSLTプロセッサーもJavaで書かれたものがいくつか公開されている。たとえばApache.orgのXalan Jumpはその1つだ。

Jump xml.apache.org



XSLTによるXML→HTML変換

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
 <xsl:template
    match="*|@*|comment()|processing-instruction()|text()">
  <xsl:copy>
   <xsl:apply-templates
     select="*|@*|comment()|processing-instruction()|text()" />
  </xsl:copy>
 </xsl:template>
</xsl:stylesheet>
```
XSLの例

W3C（ダブル・スリー・シー／World Wide Web Consortium）：WWWの発展との相互運用性の確保のために、企業や研究所、大学をメンバーとして設立された世界的な組織。HTMLやCSS、XMLの仕様書を勧告として公開している。

chap.5

Java製品を使ってみよう

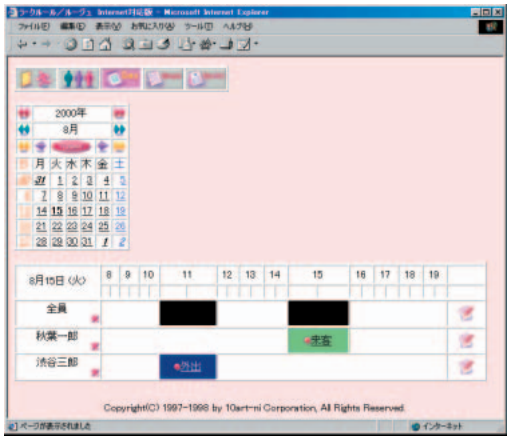
開発者の間にはJavaは定着したものの、普通のパソコンユーザーが実際にJavaを目にする機会はなかなかないのではないだろうか。最後にJavaで作られた製品を紹介しよう。ソフトがJavaで書かれることは、当たり前になってきたことがわかるだろう。

□サーブレット版グループウェア「ラ・クール／ルージュ」ー

Javaを利用した製品は数多く登場しているが、アプレットやアプリケーションなど、クライアントサイドのJavaを利用したものが多かった。サーバーサイドのJavaならインターネットやイントラネットで最大の力を発揮できる。重要なのは、アプレットやアプリケーションとサーブレットやEJBをうまく組み合わせることだ。

グループウェアであるラ・クール／ルージュ for Internet Ver.2は、まだJavaが認知されていないときから精力的にJava製品を発表しているテンアートニの製品だ。従来のラ・クール／ルージュはアプレットだったが、バージョンアップしてサーブレットを採用した（イントラネット向けにはアプレット版も使える）。サーブレットの採用によって、通信環境の充実していない場所からでも快適に利用できる。また、ポストPC時代を見据えてiモードやWorkPadからの利用も可能にしている。

機能は、個人やグループのスケジュール管理が中心で、小規模なオフィスには最適な内容だ。シンプルな構造で使い勝手よく仕上がっている。テンアートニのサイトにデモがあるので試してみよう。

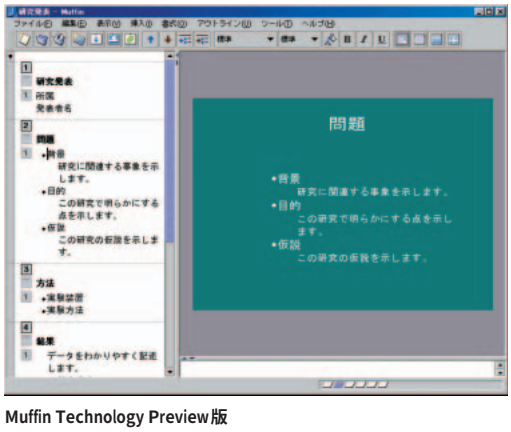


ラ・クール／ルージュ for Internet ver.2
www.10art-ni.co.jp/rouge2/

□Java & XML対応プレゼンソフト「Muffin」

一太郎とAtokでお馴染みのジャストシステムは、近年Javaへの取り組みに力を入れている。昨年末にはJava 2に完全対応したワープロ一太郎Arkを発売し、今年はJava版のプレゼンソフトMuffinと表計算ソフトChocoの開発を発表した。オフィス製品をJavaで書かれたソフトで揃えていくようだ。Muffinは現在ジャストシステムのサイトからベータ版（Technology Preview版）をダウンロードして体験できる。

これらのオフィス製品に共通しているのは、XMLを積極的に取り入れていること。一太郎ArkとMuffinは標準ファイル形式としてXHTMLを採用している。Muffinではアウトラインの編集に力点が置かれ、XMLの構造化文書という特徴を活かしたプレゼンテーションを作成できる。実際にプレゼンテーションを行うときは、WWWブラウザーで表示すればいい。また、一太郎ArkではJavaとXMLの長所である国際化機能を活用し、日本語と中国語などを混在させた文書を作成できる。インターネットの標準データ形式となったXMLと「どこでも動く」Javaの長所を活用したオフィス製品の未来を感じさせてくれる。



Muffin Technology Preview版
www.justsystem.co.jp/ark/

グループウェア：企業内ネットワークを活用して情報共有やコミュニケーションの効率化を図り、グループによる協調作業を支援するためのソフトウェアの総称。

Javaでつかむ
新しいビジネスチャンス

**Javaはインターネット上で展開されるビジネスをどう変えていこうとしているのか。
サーバー上のJavaと携帯端末のJavaで注目を集めている企業を訪れた。**

Javaの拡張性を評価 ～松井証券 Jump www.matsui.co.jp

松井証券株式会社は、今年7月に証券取引サイトのシステムを一新し、アプリケーションサーバーにJ2EE対応のBEA WebLogic Serverを採用した「ネットストック2000」を開始した。巨大なサービスのためにJavaを選んだ理由は何か、ネットストック運用部部長の日沼徹さんに事情を聞いてみた。

「今回のシステム構築の柱は4本あります。サンのStarfireとオラクルのOracle 8によるデータベースサーバーの増強、BEAのWebLogic Serverの採用、インターネットマルチフィードのハウジングの利用、Javaによるソフトウェアの刷新です。

私たちは予想していなかったトランザクションの等比級数的な増大に悩んでいました。小手先でシステムを増強するのではなく、スケーラビリティーを考えていっぺんにシステムを変えることにしました。このビジネスはインフラ商売ですから、勝つためには先手先手を打っていかないと。

WebLogicを選んだ理由は、データベース処理とHTTPという設計思想が異なるものを結び付けるミドルウェアには、実績のあるものを選ぶのが重要だからです。

“ビジネスクリティカル”と言えますが、商品のラインアップを揃えていくとき、同じやり方でシステムを作っていくのは難しい。高い拡張性を実現できるJavaは魅力的です。今後は携帯電話などのシンクライアントを前提としたシステムを作っていく必要があります。Cではソフトウェアの作りこみが必要ですが。Javaなら拡張していけばいいわけです。

今年のJavaOneでは夜11時でも多くの人が集まって、質問が飛んでいました。Javaは今それだけ注目されているということです。」


松井証券株式会社
ネットストック運用部部長　日沼 徹氏

iモード用Javaで先陣を切る ～ケイ・ラボラトリー Jump www.klab.org

iモードへのJava搭載が注目されるなかで、携帯電話向けコンテンツ開発の株式会社サイバードは、Javaコンテンツ開発事業のために、株式会社ケイ・ラボラトリーを設立した。すでにケイ・ラボラトリーは、今年のJavaOneに出展されたLG TeleComのJava搭載端末（208ページ写真）のためにサンプルプログラムを作成している。代表取締役社長である畠田善丈さんに今後の狙いについて聞いた。

「私たちの社員は10名で、全員が開発者です。今のところiモードについては、NTTドコモが仕様を公開していないので困っているところですが（注：8月8日現在）、LG TeleComの端末やMIDPを使って開発しています。

Javaが携帯電話に搭載されると、コンテンツを作る側にとっては、HTMLと違って技術的障壁が出てきます。CPUやメモリー、帯域幅に制限のある環境では、Javaの知識だけでは足りません。組み込み系のJavaのノウハウを持っている技術者の数は本当に少ない。私たちがJavaコンテンツのお手伝いをすれば、クリエイターがアイデアや想像力を活かせるようになります。オーサリングツールの提供も行う予定です。

最初に出てくるコンテンツは、ゲームでしょう。ニッチタイムのための“育てゲーム”などが有望です。データを端末側で描画すれば、高速な地図アプリケーションもできます。

コンテンツがリッチになるだけではありません。分散コンピューティングが実現する可能性があります。データをばらまいておいて、充電している間にヒトゲノムの解読をするとか。新しい社会インフラが登場することになると考えています。」


株式会社ケイ・ラボラトリー
代表取締役社長　畠田善丈氏

スケーラビリティー：ソフトウェアやコンピュータシステムにおいて、負荷が増大しても速度や安定性に影響が出ないこと。

［インターネットマガジン　バックナンバーアーカイブ］ ご利用上の注意

このPDFファイルは、株式会社インプレスR&D（株式会社インプレスから分割）が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン　バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

http://i.impressRD.jp/bn

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容（技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど）は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者（執筆者、写真の撮影者、イラストの作成者、編集部など）が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元（株式会社インプレス R&D）、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D（当時は株式会社インプレス）と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接的および間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先
株式会社インプレスR&D
All-in-One INTERNET magazine 編集部
im-info@impress.co.jp