

携帯機組み込み Java の実用化

503i では、ゲームやエージェント系サービスを実現するために Java の機能を搭載した。Java は、「Write Once, Run Anywhere」という特徴をもち、近年注目を浴びている技術である。

本稿では、503i に搭載した Java の機能について概説する。

おおせき えりこ やまだ かずひろ
大関 江利子 山田 和宏

1. まえがき

従来の i モード携帯機では、音声通話はもちろんのこと、その他にブラウザによるコンテンツの閲覧やメールの送受信、メロディのダウンロードなどが可能であった。今回、i モード携帯機に搭載された Java * 実行環境[1],[2]では、さらにプログラムのダウンロード、および実行が可能となる。ここでは、i モード携帯機に搭載された Java 実行環境とはどのようなものか、またその Java 実行環境でどのようなことが可能となるのかについて述べる。

2. Java 実行環境

Java といったときには、一般には2つの意味をもつ。1つは、プログラミング言語としての Java であり、もう1つは、実行環境としての Java である。503i では、後者を携帯機に搭載しており、Java 言語で記述されたプログラムを実行することが可能である。

Java 以外の通常のプログラムはコンパイルにより基本 OS ごとの機械語に翻訳され、実行時は基本的にその OS 上でしか動作せず、他の OS (オペレーションシステム) 上で動作させるためには別のプログラムが必要となる。VM (Virtual Machine) の概念はこの問題を解決することができる。Java 言語プログラムは、コンパイルによりどの OS 上の VM でも同じように理解できるバイトコード (Byte Code) に変換される。そのため、基本 OS が違った場合でも、OS 上に VM が組み込んであれば、Java 言語プログラムはいか

* Java : 米 Sun Microsystems 社が提唱している、ネットワークに特化したオブジェクト指向型開発環境である。

Java VMがOSの違いを吸収するため、デバイスA・Bともに同一のJavaプログラムを実行することが可能

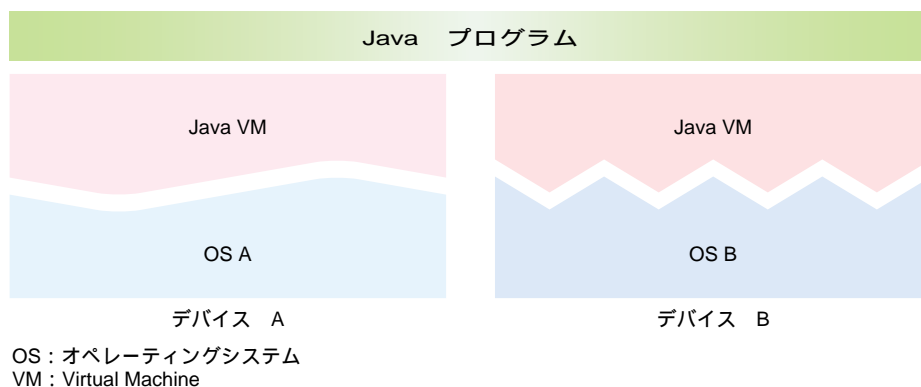


図1 Java概要図

なるOS上でも動作する。本説明内容を図示したものが図1である。

近年では小さな電子機器向けのJavaであるJ2ME (Java 2 Platform Micro Edition) が注目を浴びており、携帯情報端末 (PDA : Personal Digital Assistant) や携帯機などの携帯電子機器をはじめ、家電製品や自動車などさまざまな電子機器へVMを組み込むことが期待されている。503iにおいても、J2ME/CLDC (Connected Limited Device Configuration) を組み込んでいる。

503iにおいて、Javaプログラムはアプレットではなく、アプリケーションとして動作する。アプレットは、ブラウザ上で動くのに対し、503iでは、Javaアプリケーション単独で動作する。Javaアプリケーションのダウンロード時にはブラウザを介するが、一度ダウンロードを行えば、以降はブラウザを起動することなくJavaプログラムを実行することが可能である。すなわち、Javaプログラムは、従来のiメロディや、iアニメのダウンロード・保存と同等の操作性を実現している。

3. CLDC

CLDCは、KVM (K Virtual Machine) をベースとするConfigurationであり、CPU (Central Processing Unit) 性能やメモリ容量などに限りがあるようなネットワーク接続型の小型デバイスを対象としている。アプリケーションの上位互換性と移植性を確保するため、基本的には、上位のJava実行環境であるJ2SE (Java 2 Standard Edition) のサブセットになっているが、VMのメモリサイズの制約上、浮動小数点演算や、リフレクションなどの機能には未対応となっている。

CLDCの詳細については、米Sun Microsystems社のホームページを参考にしていきたい[1]。

4. iモード拡張API

J2MEは基本的な性能要件が類似したデバイスカテゴリごとに定義されたConfigurationと、特定の分野・業種ごとに定義されたプロファイルの組み合わせによって構成され、iモード拡張API (Application Programming Interface) は後者のプロファイルという位置付けになる。ドコモも参加しているJavaの標準化プロセスJCP (Java Community Process) を経て策定した規格としてMIDP (Mobile Information Device Profile for the J2ME Platform) というプロファイルもある。MIDPがターゲットとしている機器は双方向pager、PDA、携帯電話と広範囲となっている。一方、iモード拡張APIはiモードサービスにターゲットを絞って最適化された規格のため、MIDPとの互換性はない。

まず、503iに搭載したJava実行環境のアーキテクチャを図2に示す。図2のように、KVM・CLDCクラスライブラリ・iモード拡張ライブラリを搭載した機器であれば、同じプログラムを動作させることが可能である。iモード拡張ライブラリは、最低限の互換性を保証するべく規定したものである。ゲームなどは、画面サイズや、キーボードの特徴などを意識してプログラムを作成しなければいけないため、携帯機ごとのチューニングが必要となるかもしれないが、トランザクション系のプログラム (株価情報、天気予報など) では、同じプログラムを動作させることが可能である。以下、それぞれのコンポーネントについて説明する。なお、ここでは、APIの概要のみを説明する。詳細は、ドコモのホームページにて公開している「iモード対応Javaのスペック」を参考にしていきたい[2]。

4.1 スクラッチパッド

スクラッチパッドとは、アプリケーションからアクセス

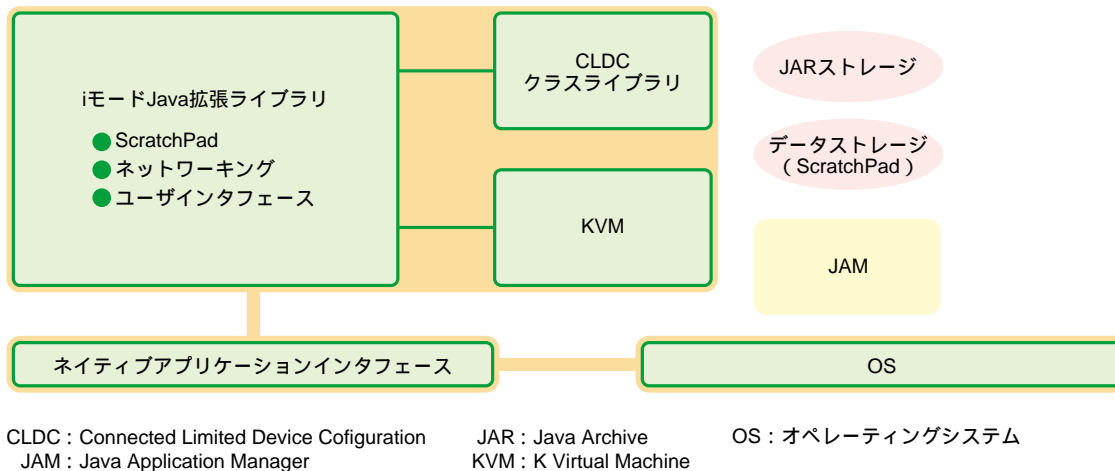


図2 iモード対応Javaのアーキテクチャ

表1 コンポーネントとHTMLとの対応

コンポーネント(Java)		HTMLのタグ(ブラウザ)	備考
文字列ラベル		<PLAINTEXT>	
画像ラベル			
ボタン		<INPUT type = " submit " >など	
テキスト入力		<INPUT type = " text " > <TEXTAREA>	1行テキスト 複数行テキスト
リスト	オプション	<SELECT> <SELECT SIZE = " 1 " >	ポップアップ型
	シングル選択	<SELECT SIZE = " 行数 " >	ボックス型
	複数選択	<SELECT SIZE = " 行数 " MULTIPLE>	行数は1以上
	チェックボックス	<INPUT type = " checkbox " >	ボックス型
ラジオボタン	<INPUT type = " radio " >		
ティッカー		<MARQUEE>	

HTML : Hyper Text Markup Language

可能なデータの保存領域であり、たとえばゲームのスコアや、各種パラメータなどを保存するために使用される。503iでは、1アプリケーションあたり、5～10kBのスクラッチパッドを使用することが可能であるが、後述するADF (Application Descriptor File) においてあらかじめ使用サイズの宣言が必要となる。また、セキュリティを確保するため、複数のアプリケーションでスクラッチパッドを共有することはできない。

4.2 ユーザインタフェース

Javaアプリケーションを実行する際のユーザインタフェースは、大きく2種類に分けられる。1つは、コンポーネントベースの高レベルAPIであり、もう1つは、ゲームなどのアプリケーションを作成するのに最適な低レベルAPIである。

高レベルAPIにおける各コンポーネントは、従来のブラウザとほぼ同等な操作性を実現するよう設計されている。

主なコンポーネントとブラウザの表示との対応を、表1に示す。

表1のようなコンポーネントを用いた場合、アプリケーション作成者の自由度は低くなるが、少ないコードで高機能なアプリケーションを作成することが可能となる。

低レベルAPIを用いると、アプリケーション作成者は、携帯機の画面サイズやキーボードの特徴などを意識してプログラミングすることができ、特にゲームのようなアプリケーションを作成する場合に有効である。したがって、アプリケーション作成者の自由度は高いが、その代わりにアプリケーション作成者がアプリケーションの挙動すべてを管理しなければならない。

また、API以外の503iの特徴として、Javaピクトというもの設けた。(写真1中の「」マーク)これは、Javaアプリケーションが実行中であることを、明示するための表示である。



写真1 Java ピクト

4.3 ネットワークの使用

503i のJava 実行環境では、Java アプリケーションがサーバと通信するためのAPIを搭載しており、HTTP (Hyper Text Transfer Protocol)、およびHTTPS (Hyper Text Transfer Protocol Security) での通信が可能である。Java アプリケーション実行中にネットワークを介して画像ファイルやメロディファイルを取得し、携帯機で再生することが可能である。また、携帯機にダウンロード可能なアプリケーション、すなわちJAR (Java Archive) ファイルのサイズは10k バイトに制限されており、さらに携帯機の処理能力も限られているため、サーバ上のCGIを用いるなどして、携帯機とサーバとで処理を分散させることも可能である。ただし、ネットワークを使用するには、後述するADFに宣言が必要となる。

5. Java 実行環境の マネージメント(JAM)

503i では、アプリケーションマネージャと呼ばれる、ダウンロードしてきたJava アプリケーションや、VMを管理するための機能を備えている。JAM (Java Application Manager) の主な機能は以下のとおりである。

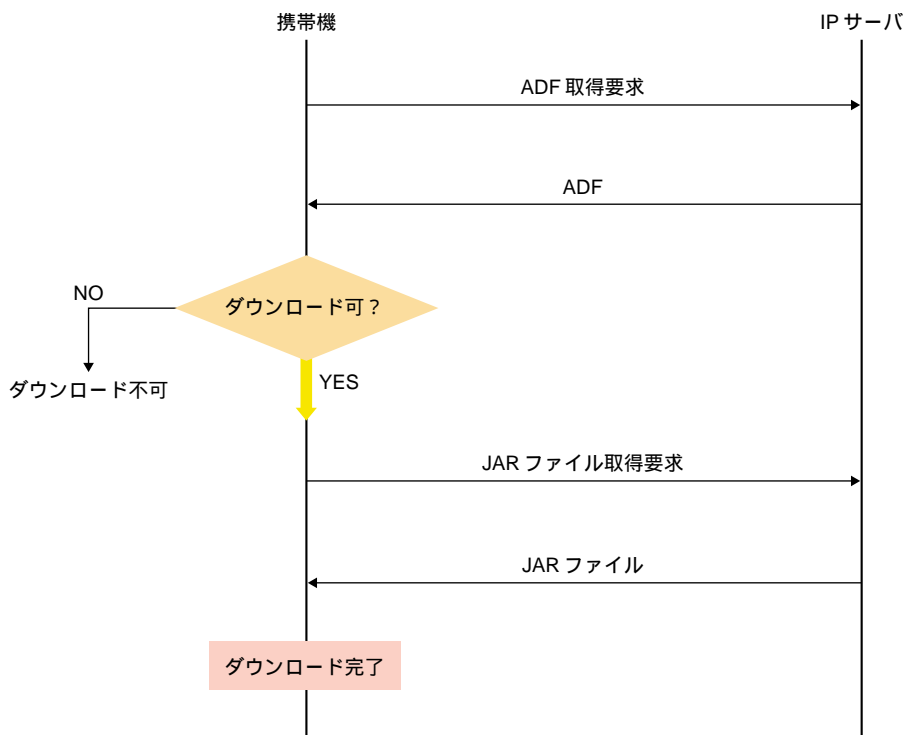
5.1 Javaアプリケーションのダウンロードと保存

HTML コンテンツ内に<OBJECT>タグが存在する場合、Java アプリケーション、JAR ファイルのダウンロードが可能となる。Java アプリケーションのダウンロードシーケンスを図3に示す。

まず、JAM はADFを取得し、そのアプリケーションが携帯機にダウンロード可能かどうかを判断する。ADFに記載されている情報を表2に示す。

5.2 Javaアプリケーションの管理

JAM は、Java アプリケーションをダウンロードする際、上述のとおり、ADFの情報に基づいてダウンロード可否の判定を行う。一例を挙げると、JAR ファイルのサイズが、



ADF : Application Descriptor File
IP : Information Provider (情報提供事業者)
JAR : Java Archive

図3 ダウンロードシーケンス

携帯機の許容するサイズ以上であれば、JAR ファイルはダウンロードされないことになる。また、ダウンロード時以外でも、JAM は ADF 設定値により、さまざまな管理を行う。詳細を下記に示す。

① 自動起動設定

ADF の “ LaunchAt ” キーで指定された時間間隔おきに、Java アプリケーションを自動起動させることが可能である。ただし、ユーザの意図しないところで、Java アプリケーションが起動してしまう可能性もあるため、本機能の有効・無効をユーザが設定できるようになっている。自動起動機能を用いたアプリケーションの一例として、株価情報や天気予報など、即時性が求められる情報を一定時間間隔で更新するようなアプリケーションが考えられる。

② ネットワーク接続設定

ADF に “ UseNetwork ” キーが指定されている場合、その Java アプリケーションは、アプリケーションのダウンロード元サーバと通信することが可能である。ただし、従来のブラウザとは異なり、アプリケーションが自発的に通信を行う（ユーザ操作なしで通信を行う）可能性もあるため、ネットワーク接続可否をユーザが設定できるようになっている。

③ バージョンアップ

ユーザがアプリケーションのバージョンアップ機能を選択すると、JAM はそのアプリケーションの ADF を再度取得し、すでに保存してある ADF の LastModified (表2) と、新規に取得した ADF の LastModified とを比較し、更新されている場合のみ JAR ファイルをダウンロードする。また、このとき既存のスクラッチパッドの内容はバージョンアップ後のアプリケーションでも使用可能である。

5.3 セキュリティチェック

JAM は、Java アプリケーションがサーバと通信しようとしたときに、そのサーバが通信可能かどうかをチェックする。503i に搭載する Java 実行環境では、セキュリティの観点から通信可能なサーバとして下記の条件を設けている。

- ① アプリケーションをダウンロードしたサイトと同ホスト・同ポート・同スキーム（プロトコル）であること。
- ② 通信処理の結果としてリダイレクションは実行しない。

表2 ADF の情報

キー値	備考
AppName	アプリケーションの名前
AppVer	付加情報
PackageURL	JARファイルの所在
AppSize	JARファイルのサイズ
KvmVer	アプリケーションの実行可能なKVMバージョン
SPsize	使用するスクラッチパッドのサイズ
AppClass	アプリケーション実行時のメインクラス
AppParam	アプリケーション実行時のパラメータ
LastModified	アプリケーションの最終更新日時
UseNetwork	ネットワーク機能の利用宣言
TargetDevice	特定機種向けのアプリケーションであることを示す
LaunchAt	自動起動の宣言

ADF: Application Descriptor File
 APP: Application
 KVM: K Virtual Machine
 SP: Scratch Pad

6. Java 実行環境におけるセキュリティ

Java 実行環境の携帯機への搭載に伴い、セキュリティ面でもさまざまな考慮を行っている。

6.1 VM レベルでのセキュリティ

J2ME/CLDC では、アプリケーションの開発時に、バイトコードの事前検証を行う。この事前検証では、通常、実行時にメモリに作成されるスタックマップなどをクラスファイルに情報記述する。この際、クラスフォーマットは JDK (Java Development Kit) などの Java の標準フォーマットはそのままに、アトリビュート領域に情報を追記しただけのものである。さらに、J2ME/CLDC では、実行時にもバイトコードの事前検証を行う。これは、コンパイル時に事前検証されたクラスファイルの情報をもとに、正しいスタック上の処理が行われるかを検証する作業である。この検証により、JDK が提供しているシステムレベルセキュリティ機能と同等の機能を CLDC は有している。

6.2 アプリケーションレベルでのセキュリティ

CLDC では、システム破壊などの悪意を持ったアプリケーションが、故意的に使用するような機能は、仕様として規定されていない。API が存在しないため、悪意を持って危険性の高いアプリケーションを書くのは不可能である。同様に、iモード拡張 API においても、危険性の高いアプリケーションが利用する可能性がある API は、仕様化していない。

また、システムリソースへのアクセスは、JAM が管理し

ており、携帯機のメモリダイヤル機能やメール機能などへのアクセスは不可能である。JAMは、KVMから独立したコンポーネントであり、JavaアプリケーションからJAMを制御することは不可能である。さらに、同時に実行可能なJavaアプリケーションは1つのみとしているため、アプリケーション同士が干渉を起したり、別のアプリケーションのデータを利用したりすることも不可能である。すなわち、アプリケーションで利用されるデータなどは、そのアプリケーションの中でのみ利用可能であり、JAMというネイティブコンポーネントがKVMやアプリケーションを管理しているため、仮に悪意のあるJavaアプリケーションが実行されても携帯機のセキュリティは保たれる。

7. あとがき

今回、携帯機にJavaを搭載したことは、新たなサービス領域を開拓するための第一歩といえよう。その本質は、外部からダウンロードしたプログラムを携帯機の中で動かすという全く新たな試みなのである。今後も携帯機のマルチメディア化をさらに進めていく予定である。

文 献

- [1] Sun Microsystems社ホームページ (<http://java.sun.com/>)
- [2] ドコモホームページ内「iモード対応Javaのスペック」 (<http://www.nttdocomo.co.jp/i/java.html>)

用 語 一 覧

ADF : Application Descriptor File	JAM : Java Application Manager
API : Application Programming Interface	JAR : Java Archive
APP : Application	JCP : Java Community Process
CLDC : Connected Limited Device Cofiguration	JDK : Java Development Kit
CPU : Central Processing Unit	KVM : K Virtual Machine
HTML : Hyper Text Markup Language	MIDP : Mobile Information Device Profile for the J2ME Platform
HTTP : Hyper Text Transfer Protocol	OS : オペレーティングシステム
HTTPS : Hyper Text Transfer Protocol Security	PDA : Personal Digital Assistant (携帯情報端末)
IP : Information Provider (情報提供事業者)	SP : Scratch Pad
J2ME : Java 2 Platform Micro Edition	VM : Virtual Machine
J2SE : Java 2 Standard Edition	バイトコード : Byte Code