PDAとJava

Dec. 19, 2000
ver.0.47

シャープ株式会社
情報家電開発本部
情報通信システム開発研究所
生駒 孝夫
ikoma@isl.nara.sharp.co.jp

IAJ/'Java & Mobile' Winter Seminar in IW2000

PDAとJava

- □PDAと情報家電
- □Java2 Platform, Micro Edition
- □PersonalJava
- □Zaurus上のPersonalJava
- □Zaurusへの移植
- □JiniとZaurus

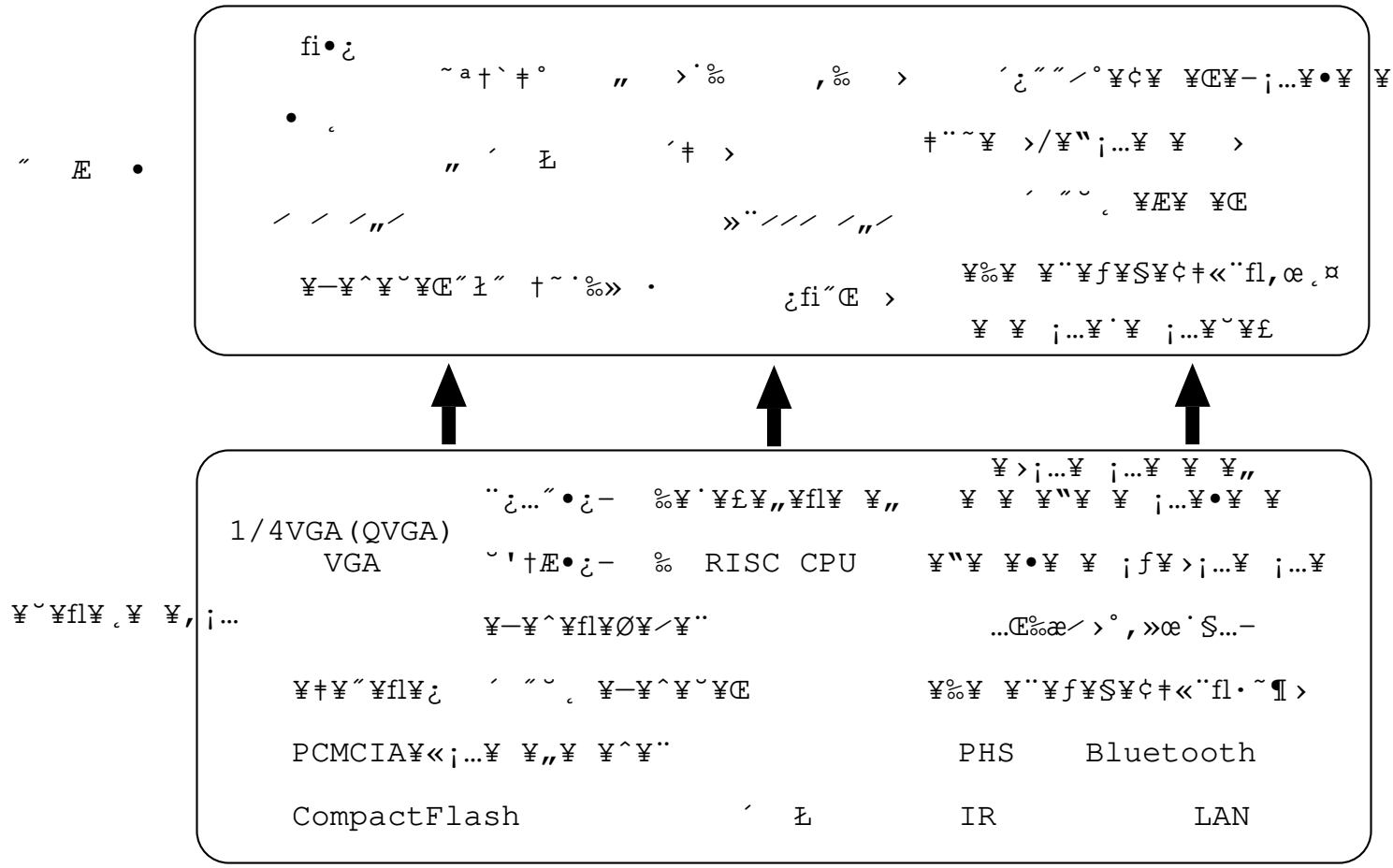
PDA

- Personal Digital Assistant
  - 1992(?): John Scully がNewtonテクノロジー構想で発表
- 携帯情報端末
- オーガナイザ、電子手帳との違い
  - プラットフォーム、汎用性
  - コミュニケーション
- 個人用途と業務用途
- $150-$1500
  - スイートスポットは$300くらい
  - 小さいから安い?

PDAの要求条件

- 価格
  - 部品コスト
- モバイル
  - フォーム・ファクタ、重量
    - シャツのポケットに入る(?)
    - ペン・オペレーション(キーボード、マウスなし)
  - 電池駆動
    - 省メモリ、ディスクレス
  - コネクティビティ(ワイアレス)
    - 間歇接続
    - より高帯域へ
- 視認性
  - 反射型液晶と透過型液晶、バックライト
- アップグレード、バグフィックス
  - パッチ機構、アプリケーション更新

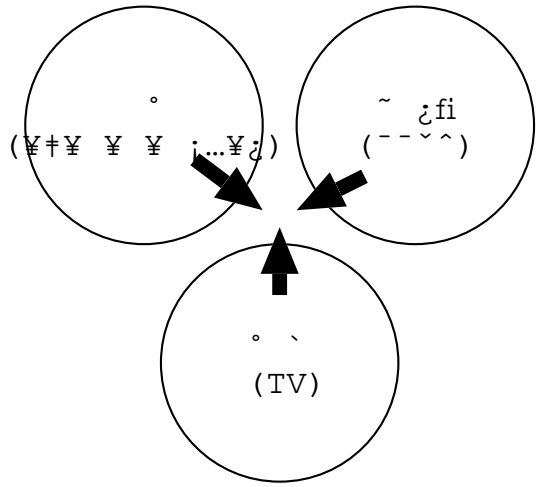
トレードオフ



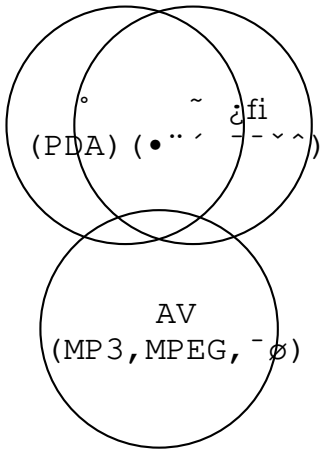
PDAのアプリケーション

- □PIMアプリケーション(Personal Information Manager)
  - ○アドレス帳(電話帳)、スケジュール、ToDoリスト
- □インターネット
  - ○メール、WWW
- □PC連携、同期
- □業務用アプリケーション
  - ○営業員支援、在庫情報、顧客情報、旅費交通費精算
  - ○プレゼンテーション
  - ○事故車、建築現場確認(カメラ + メール)
- □Eコマース(電子商取引)、電子財布
- □地図、ナビゲーション
- □エンターテインメント
  - ○ゲーム、音楽、動画、...

コンバージェンス



Convergence



Convergence(PDA)

PDAと電話との融合へ

- □オーバラップするアプリケーション
  - ○PIM(スケジュール、アドレス帳(電話帳))
  - ○時計
  - ○メール、WWWブラウザ
  - ○メモ、音声メモ
- □次はAV
  - ○ディジタル・カメラ
  - ○MP3
  - ○MPEG4
- □携帯電話とPDA、両方もつ必要性はあるか?
  - ○電話付PDA = 大型液晶付電話(?)
  - ○テレビも見られる、音楽も聞ける(?)
- □電話は外づけ(2box) or 組込み(1box)？

PDAと携帯電話の棲み分け

- □携帯電話があるので、PDAもつのをやめた?
- □画面の大きさ
  - ○ビューアとしての機能
    - ▹携帯電話の画面で地図はみづらい
  - ○大きな液晶を耳に当ててもしかたがない
- □入力手段
  - ○テンキーでメール打てるのは30才まで(?)
  - ○かといって、手書きで大量/高速入力には無理がある
- □電話より高機能なJava
  - ○携帯電話と同じレベルでは存在意義はない

PDAのオープン化

- □PDAの進化
  - ○組込みアプリケーションのみ(専用システム)
  - ○カスタム化可能(OEM、特定バーティカル市場向け)
  - ○オープンプラットフォーム
- □オープン性---プラットフォームとしてのPDA
  - ○コンシューマ・プロダクツ vs プラットフォーム
  - ○すべてのアプリケーションをPDAベンダーが用意できない
  - ○オープンなコンシューマ・プロダクツ(?)
- □ソフトウェア(アプリケーション)開発者のサポート
  - ○開発環境、ツール
  - ○技術情報公開、情報交換の場、ソフト流通の場
  - ○継続性
- □Javaの搭載
- □オープン・ソースPDA(!?)
  - ○組込みLinux

「組込み機器」

- 「エンベデッド」と「embedded」の違い
- 日本
  - 組込み機器 = コンシューマ機器
    - 携帯、電池駆動、低価格、...
    - PDA、ディジタルカメラ、STB、...
- 米国
  - 組込み機器
    - 交換機、計測機器、ルーター、ミサイル(?)、...
- EmbeddedXXXをそのまま日本へもってきても...!?

Javaとオープンソース

- SCSLはオープンソース・ライセンスではない
  - Open Source Initiativeの定義に該当しない
  - ソースコードは無償で利用可能
- Javaコミュニティとオープンソース(Linux)コミュニティとの微妙な関係
  - JavaはGPL化されるか?
  - 仕様/互換性を骨抜きにされないために?
- Javaコミュニティ・プロセス(JCP)
- Jiniコミュニティのこころみ
  - http://www.jini.org/

SCSLライセンス・モデル

- Sun Community Source License
- オープン・ソースとビジネス・モデルの両立の試み
- 研究開発、社内利用には無償でソースコード公開
  - オンライン・アグリーメント
  - プロトタイプ開発までは費用発生せず
- 商用ライセンスで費用発生
  - 互換性テストなど
- J2SE、Jini、PersonalJava、PicoJavaコア、等
- ライセンシー間でのソース・コード交換可能

Java Community Process(JCP)

- Java技術関連の仕様を公平・迅速に開発するための公式プロセス
  - 業界の少数の専門家グループによる仕様案
    - Sunのライセンシーでなくても仕様開発に参加可
    - Sunの技術者の参加は任意
  - 仕様とともに、参照実装、互換性テスト・スイートの開発を義務化
  - 手続きのルール化
    - 独立監査機関による標準化手続きのレビューと結果公表
  - プロセス
    - 新仕様の提案 -- JSR(Java Specification Request)
    - エキスパート・グループの募集 -- CAFE(Call for Expert)
    - エキスパート・グループによる仕様案の作成、レビュー、改訂
    - 公開レビュー
    - 初版公開
    - 仕様のメンテナンス
  - http://java.sun.com/aboutJava/communityprocess/

Javaと情報家電

- PDA、STB用の言語として開発開始(1991頃)
  - *7(Star Seven)
- インターネットのブームに乗ってデビュー(1995)
  - アプレットの強調
- デスクトップでの失望(1998頃)
  - 過剰な期待への反作用
  - 互換性、性能に問題
- イントラネット、サーバ系で実用化先行(1997-1999)
  - メモリ、ネットワークのリソースと性能の制約すくない
  - 互換性に問題の多かったGUIのウェイト軽い
- 再び情報家電へ(1999-)

情報家電とは?

- Information Appliance
- Convergence(融合)が家庭へも
  - コンピュータ(PC)・通信(電話)・放送(TV)
- 家電の情報化とコンピュータの家電化
- コンシューマ、一般ユーザ(非専門家)が利用
  - 必ずしも家庭での利用に限定されるわけではない
- 組込み機器の一分野
  - 情報家電
    - ワープロ、PDA、カーナビ、ディジタル・カメラ
    - 電話(携帯電話、ウェブフォン)、FAX
    - AV機器(TV、ビデオ、MD、MP3、...)
    - 白モノ(冷蔵庫、電子レンジ、皿洗い機、エアコン、...)
    - (家庭用)プリンタ、スキャナ
    - ペット・ロボット(?)
  - OA機器、計測機器、自動車、FA機器、自動販売機、兵器、等

情報家電の誕生

- □家電機器の情報化
  - ○ソフトウェアの大規模化
  - ○ディジタル化
- □情報機器(PC)の家電化
  - ○ユーザ層の拡大
    - ▹デザイン重視、使いやすさ
    - ▹「箱を開けたらすぐ『インターネットできる』」
  - ○低価格化
    - ▹"$500PC"
  - ○マルチメディア化
    - ▹PCでDVDを見る

コンシューマ・ユーザ

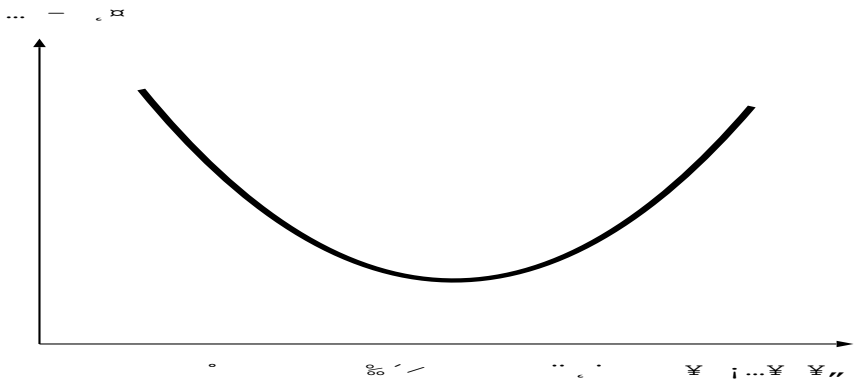
- □使う側の論理
  - ○実用性重視
    - ▹使いやすさ、安全性、ロバストネス、省スペース、デザイン
  - ○値段が手ごろ(値ごろ感)
    - ▹ランニング・コストも(省エネ)
  - ○ややこしいものはいらない
    - ▹複雑な操作には拒否反応
    - ▹とくに、とっつきやすさ(店頭での印象)
  - ○正しく動いて当然
    - ▹PCユーザほど寛大でない(青画面、爆弾)
  - ○忍耐強くない
    - ▹即時かつ継続的なレスポンス
  - ○技術者の予期せぬオペレーション(高齢者、子供も使う)
  - ○HOWは問わない

家電の発想

- □実用性と価格が最優先、余分なものは要らない
- □専用(単機能)システム
  - ○汎用性、拡張性よりも価格、使い勝手、信頼性/ロバストネスが優先
- □償却台数多い → コストのうち固定費より変動費重視
  - ○部品コスト(変動費)の最小化---CPU、メモリ、入出力デバイス
  - ○要求： 部品は安く、性能は高く
    - ▹アセンブリ言語、C言語が中心
- □売り切り
  - ○ソフトのインストールは工場出荷時(設置時)のみのケースが多い
  - ○ソフトウェアはROMベース
- □実行環境のみで十分
  - ○開発環境は別(PCのクロス開発環境)
- □メーカーは市場競争に勝ち残るためにHOWを考える

新しいビジネス・モデル

- □ハードウェアは無料または超安価で配布
- □サービスやコンテンツで課金
  - ○携帯電話、無料PC、等
  - ○FAXを貸し出す学習塾、会員にPDAを貸与する証券会社
- □投資するところと回収するところは別
  - ○トータルで利益を出す
- □スマイル・カーブ



トレンド

- □価値観の多様化(ユーザの要求の多様化)
  - ○マスカスタマイゼーション
- □パーソナル化(「個電」)
  - ○一家に一台からひとり一台へ(さらに、ひとりn台へ?)
- □モバイル化
  - ○省エネ化の要求
- □ハードウェアの高性能化/低価格化
- □ライフサイクルの短縮
- □ディジタル化/高帯域化
- □ネットワーク化
- □インタラクティブ化
- □環境重視、省エネ化、家電リサイクル
- □高機能化/多機能化/複雑化

組込み機器のソフトウェア危機

- □組込み機器のソフトウェアはますます複雑、大規模に
  - ○ユーザニーズの多様化
  - ○要求される機能の複雑化
  - ○ハードウェア・コストの低下
- □バグの影響はとてつもなく大きい
  - ○正しく動くのがあたりまえ
  - ○信頼性
- □製品サイクル、開発期間の短縮
- □ソフトウェア危機
  - ○メインフレーム(70年代) → PC(80年代) → 組込み機器(90年代)
- □ソフトウェア工学の成果を組込み機器にも
  - ○もう「職人芸」では間に合わない!
    - ▹局所的最適化の合成が全域的最適化にならない
  - ○従来高価すぎたソフトウェア技術も採算が合うようになってきた

Java!? -- コンシューマには直接関係ない

- □開発者サイドのソリューション
  - ○コンシューマにとって大事なのは売値で、原価ではない
- □開発者のメリット(間接的にエンドユーザにも影響)
  - ○ソフトウェアの生産性、信頼性、開発効率
  - ○機能、つかいやすさ
  - ○コスト
  - ○拡張性、再利用/流用の可能性(融通が効く)
- □デメリット
  - ○パフォーマンス
  - ○メモリサイズ → コスト
  - ○既存のコード資産
- □メリットとデメリットのクロスする時期

一般ユーザにとってのメリット

- □間接的なユーザのメリット
  - ○アプリケーションの選択肢
    - ▹たくさんのアプリケーションが選択できる
    - ▹肌理細かなカスタマイゼーション
      設定が簡単
      いろんなものとつながる
    - ▹早くマーケットに登場する
  - ○バグがない(あたりまえ!)
    - ▹落ちない、こけない、凍らない
      (Winよりは...)
  - ○ウィルスに強い(たぶん)
    - ▹実装のバグはあるかもしれないが、モデルはしっかりしている
  - ○価格が安い
- □Sunのロゴはあってもなくてもよい

従来のJavaプラットフォームの種類

- JDK/JAE
  - Java Development Kit(開発環境)
  - Java Application Environment(実行環境)
- PersonalJava
  - アプレットのサポート
- EmbeddedJava
- JavaCard

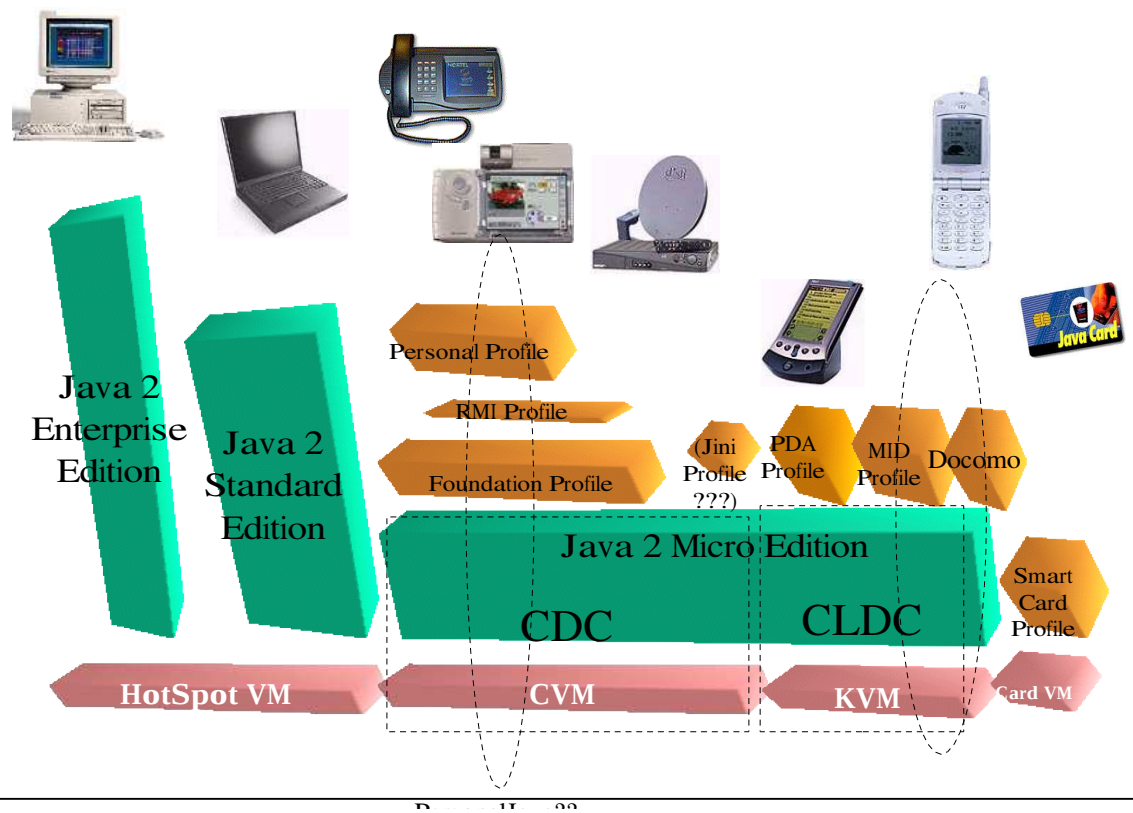
Javaプラットフォームの再編成

(1999年6月

)

- Java2 Platform, Enterprise Edition (J2EE)
  - サーバー
- Java2 Platform, Standrad Edition (J2SE)
  - デスクトップ
  - ← JDK
- Java2 Platform, Micro Edition (J2ME)
  - 組込み機器、情報家電
  - ← PJava, EJava
- JavaCard
  - スマートカード

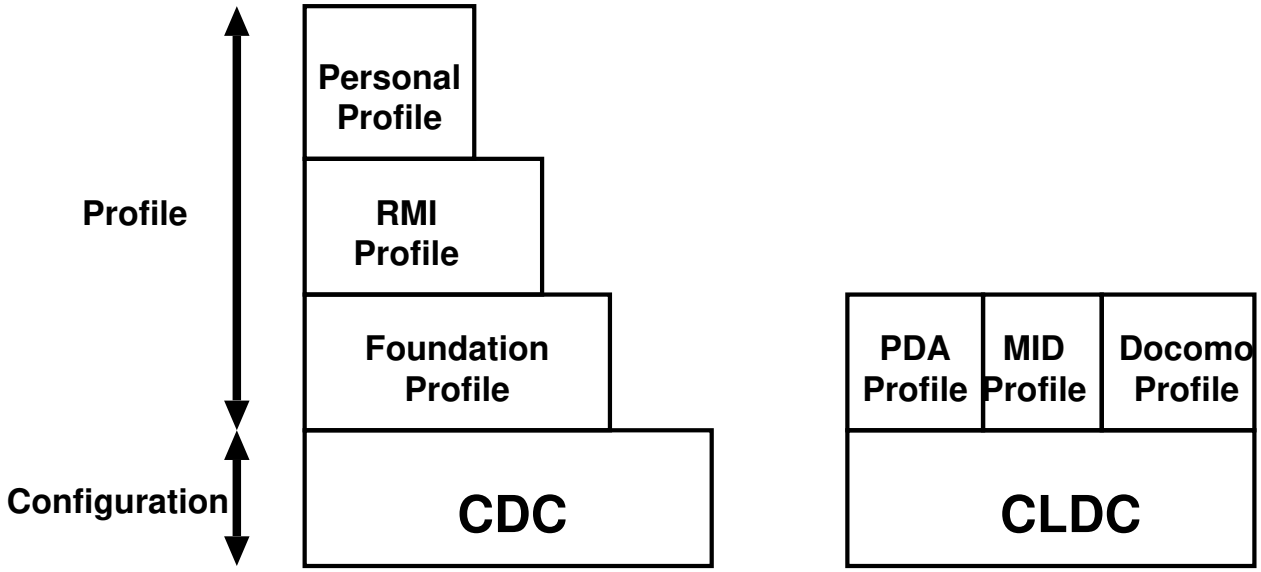
Javaプラットフォーム



コンフィギュレーションとプロファイル

- J2MEで導入された概念
- コンフィギュレーション
  - JVMと「コアAPI」
  - 広範なクラスのデバイスのためのプラットフォーム
  - CLDCとCDCのふたつ
- プロファイル
  - 特定バーティカル市場(業界、商品カテゴリ)向けAPI群
  - 下位のコンフィギュレーション上に規定
  - CLDC上のプロファイル
    - MIDプロファイル
    - PDAプロファイル
  - CDC上のプロファイル
    - Foundationプロファイル
    - RMIプロファイル
    - Personalプロファイル
  - プロファイルはネストしてよい

コンフィギュレーションとプロファイル



J2ME関連JSRステータス

- Java APIs for Bluetooth
  - (JSR82) expert group formed(11/09/00)
- PDA Profile for 2ME
  - (JSR75) expert group formed(9/14/00)
- J2ME Platform Specification
  - (JSR68) expert group formed(7/29/00)
- J2ME RMI Profile
  - (JSR66) public review closed(11/6/00)
- J2ME Personal Profile
  - (JSR62) expert group formed(7/29/00)
- J2ME Foundation Profile
  - (JSR46) proposed final draft(8/30/00)
- Mobile Information Device Profile for J2ME
  - (JSR37) final(9/19/00)
- J2ME Connected Device Configuration
  - (JSR36) proposed final draft(8/30/00)
- J2ME Connected, Limited Device Configuration(CLDC)
  - (JSR30) final(05/30/00)

ふたつのコンフィギュレーション

- Connected Limited Device Configuration (CLDC)
  - KVMベース
  - 携帯電話、PalmPilot
  - MID Profile
  - PDA Profile
- Connected Device Configuration (CDC)
  - 従来VMベース
  - ハイエンドPDA、ウェブフォン、ハイエンドSTB、...
  - Foundation Profile
  - RMI Profile
  - Personal Profile

CLDC

- J2ME Connected, Limited Device Configuration
- J2MEのコンフィギュレーションのひとつを定義
  - KVMベース
  - 小規模
  - リソースに制約
  - ネットワークに接続される機器
    - 合計128KBないし512KBのメモリ
      <= 256KB ROM/Flashかつ<= 256KB RAM
      一般には、RAMやフラッシュよりROMの方が大きい
    - 限定された電力供給(通常は電池駆動)
    - なんらかのネットワークへの接続
      間欠接続、限定された帯域幅(9600bps以下)
    - さまざまな程度のUI(あるいはUIなし)
- 対象機器: 双方向ページャ、携帯電話、PDA、POS端末、等

KVM(K Virtual Machine)

- □1999年6月JavaOneで発表
- □バイトコード実行環境
  - ○Java言語仕様のサブセット化
- □最小構成
  - ○16bit(16MHz以上)
  - ○160KB-512KB(128KB不揮発性メモリ、32K揮発性メモリ)
    - ▹KVM自身は最小40KB程度
- □移植性
  - ○Palm Pilot, Motorola双方向ページャ、次世代i-mode端末でプロトタイプ
  - ○その他、POS端末、STBなど

CLDC

- □CLDCには含まれない仕様
  - ○アプリケーションマネージャ
    - ▹アプリケーションのインストール、起動、削除
  - ○ユーザインターフェイス
  - ○イベントハンドリング
  - ○ハイレベルなアプリケーションモデル
- □CLDCで未対応の事項(Java2SEと比較)
  - ○浮動小数点(バイトコード未対応)
  - ○JNI(Java Native Interface)
    - ▹native関数は使用できない
      サンドボックスモデル、メモリ容量
  - ○ユーザ定義クラスローダ
    - ▹ビルトインのクラスローダをユーザで変更できない
  - ○リフレクション
    - ▹リフレクションの機能を利用するものも利用できない
  - ○スレッドグループ、デーモンスレッド
  - ○ファイナライゼーション
  - ○ウィークリファレンス

CLDC

□CLDCの制限事項

○セキュリティ(サンドボックスモデルを採用)

- ▹Javaクラスファイルは事前にベリファイされている
  有効なJavaアプリが保証されている
- ▹デバイス外で事前にベリファイし、スタックマップという新しい属性をクラスファイルに付加
- ▹実行時にこの属性をベリファイ
  (この属性はJ2SEのVMでは無視されるので、完全な上位互換)
- ▹アプリケーションプログラマはプロファイルとライセンシーオープンクラスで既に定義されたJavaAPIだけ使用できる。
- ▹Javaアプリケーションのダウンロードと管理はnativeコードレベルで行う(ユーザ定義クラスローダは使用できない)
- ▹VMにアクセス可能なnative関数は使用不可
  アプリケーションプログラマはnative機能を含んだ新しいライブラリをダウンロードできない
  でないと、既に提供されているJavaライブラリに含まれない任意のnative関数にアクセスできる
- ▹クラスファイルのルックアップ順序は変更不可

CLDC

□CLDCの制限事項(続き)

○エラーハンドリング

- ▹Exceptionクラスはサポートしているが Errorクラスは制限
  エラー状態からのリカバリが組込みシステムではデバイス依存
  エラーハンドリングは負荷が高く、オーバヘッドが大きい

○国際化

- ▹Reader/Writer関連はInputStreamReaderとOutputStreamWriterで実現
- ▹ローカライゼーションはサポートしない

○プロパティ

- ▹Propertiesはサポートしない
  一部のプロパティはSystem.getPropertyで対応

□追加された項目

○ROM化可能

○Connectorクラス

- ▹HTTP、ソケット、シリアル、ファイルなどすべてのコネクションを行う

MIDプロファイル

- Mobile Information Device Profile for J2ME
- CLDC上のプロファイル
- プロセッサパワー、メモリ、接続性、ディスプレイサイズに制限
- 最低限必要とされるスペック
  - [ハードウェア]
    - スクリーンサイズ96x54
    - ディスプレイデプス1bit、ピクセル形状ほぼ1:1
    - 一つ以上の入力機構(片手or両手操作のキーボード、タッチスクリーン)
    - 128KB不揮発性メモリ(MIDP)、8KB不揮発性メモリ(アプリケーション用の永続データ)、32KB揮発性メモリ(Javaランタイムメモリ)
    - 制限された帯域のネットワーク接続
  - [ソフトウェア]
    - 不揮発性メモリへのリード / ライト
    - ワイアレスネットワークへのアクセス
    - タイムスタンプに使用するための時間機構
    - ビットマップグラフィックスディスプレイへの書き込みと入力機構
    - アプリケーションのライフサイクルを管理する機構

MIDプロファイル(続き)

- MIDPに含まれない仕様
  - システムレベルAPI
  - アプリケーション管理
  - 低レベルセキュリティ
  - アプリケーションレベルセキュリティ
  - End-to-endセキュリティ
- 2つのカテゴリのAPI
  - MIDP API
    - この仕様書で定義されたAPI
  - OEM-specific API
    - デバイス固有な機能にアクセスするためにOEMで提供されるAPI

特徴的なMIDP API

□タイマー
- ○遅延やスケジュールを必要とするアプリケーション用に用意
- ○一回のみの実行と定期的な繰り返し実行のための機能を提供

□ネットワーキング
- ○HTTPプロトコル(HTTP1.1)のサブセットをサポート
- ○IPプロトコルも、ゲートウェイを経由したnon-IPプロトコルも可

□永続ストレージ(Persistent Storage)
- ○永続的にデータ格納、取り出しを行う機構を提供
- ○レコード管理システム。単純なレコード指向データベース・モデル

□アプリケーション
- ○「MIDlet」
  - ▹MIDPアプリケーションのためのアプリケーション実行環境を提供
  - ▹アプリケーションの内容記述、クラスファイル、リソースファイルから構成
  - ▹一つのJARファイルにパッケージされている
  - ▹アプリケーション管理でこのJARファイルを展開、インストールすることでMIDletの実行が可能になる

特徴的なMIDP API(続き)

□ユーザインターフェイス
- ○ディスプレイに描画するためのAPI
- ○リスト、テキストボックス、フォーム等のUIコンポーネント
- ○フォームを管理するようなスクリーン
- ○入力デバイスからの入力の受け渡しなどを行うAPIを提供

PDAプロファイル

- CLDC上のプロファイル
- GUIとデータ・ストレッジのAPIの仕様化をめざす
  - AWTのサブセット
  - 簡単な永続的記憶
    - アプリケーション、データ、設定情報
- 想定する機器
  - 低消費電力(電池駆動)
  - 512KB以上16MB以下(Javaラインタイム+ライブラリ)
  - 20,000ピクセル以上
- MIDP以上、JavaPhone未満
- PalmPilot上のプロトタイプが出発点？
- (JSR75) expert group formed(9/14/00)

CDC

- J2ME Connected Device Configuration
- Classic VMベース
- J2MEコンフィギュレーションのひとつを定義
  - 最低512KB ROM
  - 最低256KB RAM
  - なんらかのネットワーク接続
  - JVMS(2nd edition)に従ったJVMのサポート
  - さまざまな程度のUI(あるいはUIなし)
- 対象機器: STB、ウェブフォン、カーナビ、等

Foundationプロファイル

- PersonalJavaのGUIを除く部分
  - リッチなネットワークを利用できるがGUIを必要としないJava環境
  - GUIや他の機能を追加することによって他のプロファイルを構築するためのベースとなる
  - 以下のような特徴を持つ機器でJ2MEのプロファイルを提供する
    - 最低1024KB ROM(アプリケーションのメモリ要求は別)
    - 最低512KB RAM(アプリケーションのメモリ要求は別)
    - なんらかのネットワークへの接続
  - 付加的なプロファイルによって提供されない限りGUIを持たない

RMIプロファイル

- CDC、Foundationプロファイル上のプロファイル
- J2SE RMIとの互換性
- ターゲットデバイス
  - 2.5MB ROM/1.0MB RAM
    - (アプリケーションとローカライゼーションは別)
  - TCP/IP接続
  - CDC/Foundation Profile

Personalプロファイル

- 次世代のPersonalJava環境
  - 1.1.x/1.2.xのPersonalJavaアプリケーション環境との互換性
- ターゲットデバイス
  - 2.5MB ROM/1.0MB RAM
    - (アプリケーションとローカライゼーションは別)
  - なんらかのしっかりしたネットワーク
  - WWW表示やApplet実行できるレベルのGUI
  - CDC/Foundation Profile

PersonalJavaからPersonalプロファイルへ

- 現在
  - PJava仕様1.1.3 = PJAE(参照実装)3.0.2 = JDK1.1.8ベース
- 次版
  - PJava仕様1.2 = PJAE(参照実装)3.1 = Java2一部取り込み
- 将来
  - CDC + Foundation P + RMI P + PersonalP(GUI)
  - すべての機器がRMIやGUIを必要とするわけではない

PersonalJavaアプリケーション環境仕様

- 言語仕様、VM仕様はそのまま
- Required Packages
  - java.applet
  - java.awt.datatransfer
  - java.awt.event
  - java.awt.image
  - java.beans
  - java.lang
  - java.lang.reflect
  - java.net
  - java.text
  - java.util
- Modified Packages
  - java.awt
  - java.awt.peer
  - java.io
  - java.text.resources
  - java.util.zip

PersonalJavaアプリケーション環境仕様

- Optional Packages
  - java.math
  - java.rmi
  - java.rmi.dgc
  - java.rmi.registry
  - java.rmi.server
  - java.security
  - java.security.interfaces
  - java.sql
- Unsupported Pacages
  - java.security.acl
- PJAE Specific APIs
  - Double Buffering
  - Input Behaviors for Mouse-less Environment
  - Timer API
- Unsupported Class/ Unsupported Method
  - 互換性のためメソッドは存在
  - java.lang.UnsupportedOperationException

PersonalJavaのAWT

- JDK1.1ベース
- Swingは使えない
- オプショナル
  - Windowはroot windowのみ
  - Scrollbarは非サポート
    - 代わりにスクロール・ボタンを使う
  - Dialogはモーダルのみ(表示は一つのみ)
    - モードレス・ダイアログはない
  - FileDialog
    - ユーザから見えるファイルシステムがなければ不要
    - ファイルシステムのない(を見せない)機器もある
  - PopupMenu/MenuBar等はサポートしない

PersonalJava

- マウスレス環境のサポート
  - 家電には(ふつう)キーボードがない
  - ポインティングデバイスもない(?)
  - リモコン方式、十字ボタン、携帯電話式、タッチパネル、ボタン、ダイヤル
  - ディスプレイもないかもしれない、1行かもしれない
- カレント・コンポーネントの選択
  - input preference interfaces
    - NoInputPreferred
    - KeyboardInputPreferred
    - ActionInputPreferred
    - PositionInputPreferred
- Timer APIs

Truffle Graphical Toolkit

- □Peer SetのJavaによる実装
  - ○Solarisの実装ではPeer SetとしてMotifを利用
    - ▹情報家電機器には高度なウィンドウシステムはない
  - ○低レベルのグラフィクス・プリミティブの上に移植できる
  - ○通常、コンシューマ機器には高度なGUIライブラリは用意されていない
  - ○性能チューニングの余地
- □別のLook & Feelを提供するためのフレームワーク

単純なユーザ・インタフェース

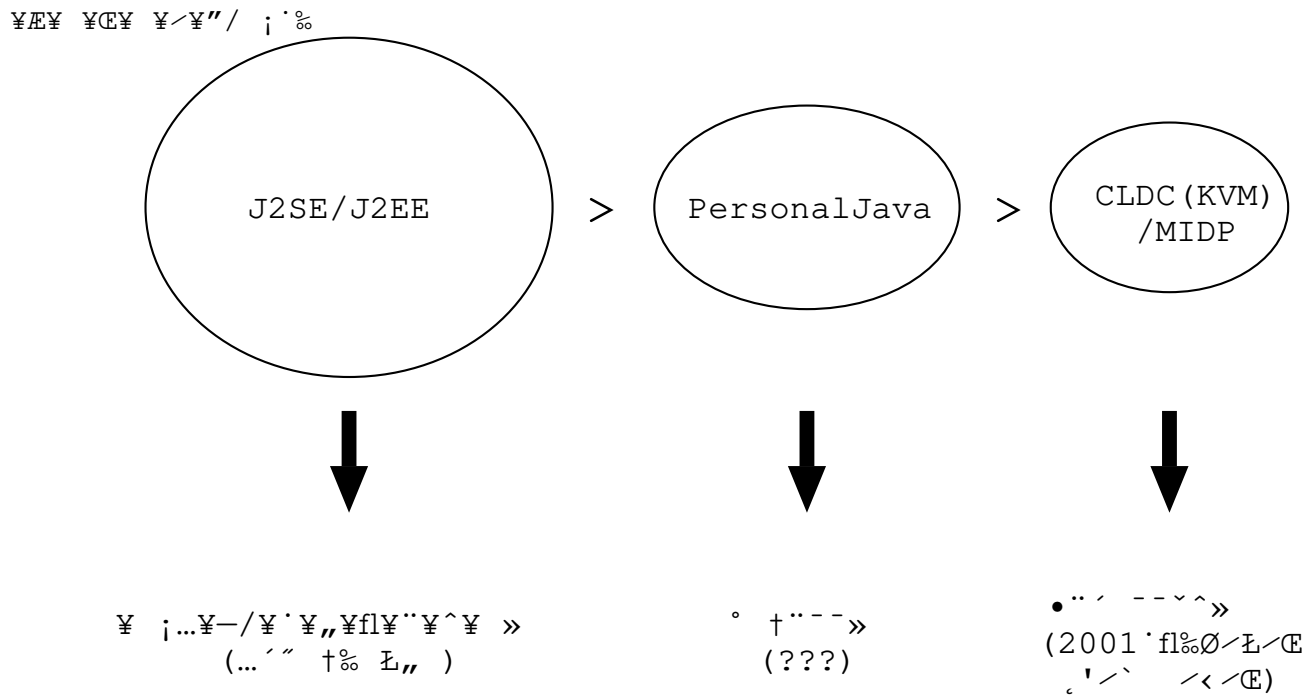
- □(ガイドライン)
- □シングル・クリック・モデル
  - ○× ダブル・クリック
  - ○× ドラッグ & ドロップ
- □GUI部品数を最小に
  - ○× オーバラッピング・ウィンドウ
  - ○× 階層メニュー

入力に対する即時/継続的なフィードバック

ROM化

- □バイトコードの自己書き換え
  - ○quick命令
  - ○参照の解決
- □マルチクラス・ファイル
- □その他のリソース
  - ○HTML、イメージ、音声、...
- □開発ツール
  - ○JavaCodeCompact

Javaのマーケット



ZaurusになぜPersonalJavaなのか

- □デスクトップのJavaと互換性が高く、既存のプログラムを利用、流用しやすい
- □数MB(8-32MB)のメモリで動作する
- □言語仕様上、プログラムが異常動作しにくい
- □基幹業務ネットワークやインターネットへの柔軟なモバイルアクセスを実現

Zaurusで動作するJava

- □PersonalJava実行環境3.0.2版
  - ○PersonalJava仕様1.1.3版(JDK1.1.6をベース)
  - ○Truffle(GUIツールキット)
  - ○Touchable(Look & Feel)
  - ○オプション
  - ○一部、制限事項
  - ○3.1版(仕様1.2版)はスキップ
    - ▹バグ対応のみバック・ポート
- □ザウルス固有機能

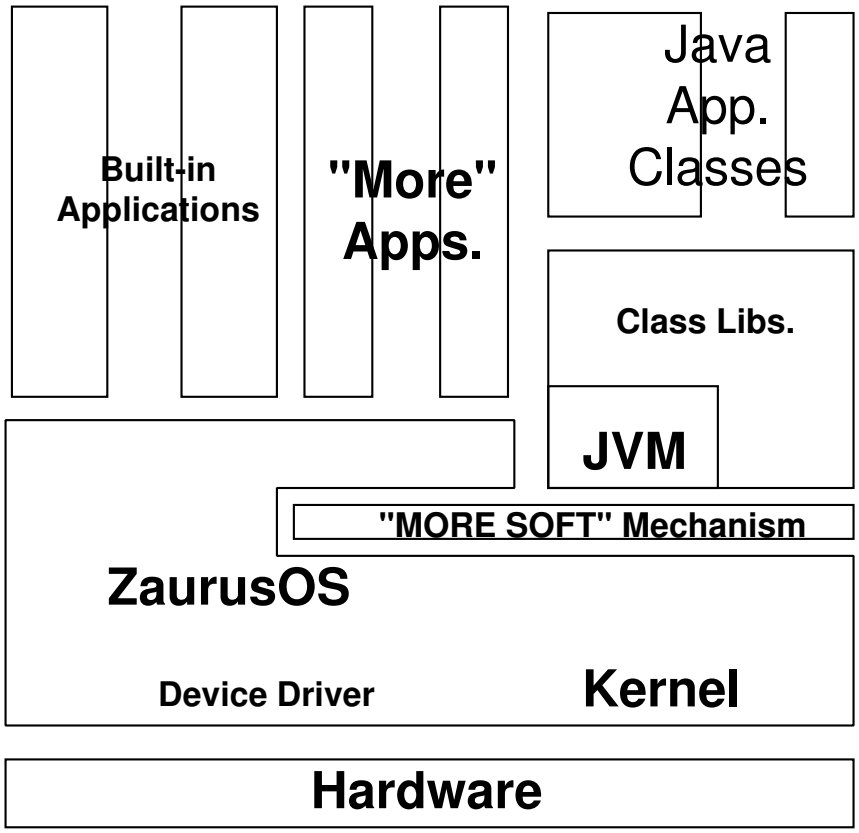
Javaが動作するZaurus

- 8MB以上RAM搭載モデル(現行4モデル)
  - MI-EX1(ICRUISE -- VGAモデル)
  - MI-C1
  - BI-L10(ビジネスザウルス、モノクロ)
  - MI-E1(予定)
  - 今後発表されるハイエンドモデル(予定)
    - (搭載RAM8MB以上)
- Javaの動作しないモデル
  - 上記以外の既存モデル
  - 搭載RAMが8MB未満のローエンド・モデル

ZaurusからみたPersonalJavaの位置づけ

- Zaurus上のひとつのアプリケーション(「MOREソフト」)としてPJVMが走る
  - 複数Javaアプリケーションの同時実行は不可
- AppletはAppletViewerでのみ動作可
  - WWWブラウザとの連携は未サポート
    - メモリ容量の問題

PersonalJava on Zaurus



ザウルスでのJavaの起動

- 他の「MOREソフト」と同様、インデックス画面からPJVMを起動
  - アイコンをクリック
- コマンドラインがない！
- 標準出力/エラー出力をどうする？
- ネットワークの接続は？

起動時オプション

- □コマンドライン引数と環境変数
  - ○Java起動用パラメータファイル
- □標準出力/標準エラー出力
  - ○ファイル and/or 画面へ出力するオプション(チェックボックス)
- □ネットワーク接続
  - ○PPP接続オプション(チェックボックス)
  - ○設定はザウルスOSの機能を利用
    - ▹ブラウザ、メール利用と同様
    - ▹接続先選択と各接続先設定
  - ○PPPダイアリングAPIも実装(予定)

Java起動用パラメータファイル

- □JAVAARGS.TXT(コマンドライン引数)

```
-vmheap2500k -ms1500k -mx1500k -oss10k sun.applet.AppletViewer DUKE.HTM
```

- □JAVAENVS.TXT(環境変数)

```
CLASSPATH=F1:JAVACLS.ZIP;F1:JAVAEXT.ZIP;F1:DUKE.JAR
HOME=F1:
JAVA_HOME=F1:
USER=ikoma
```

ザウルスでのJavaアプリケーションの登録、起動

- ローンチャ
  - アプリ名: (例: Tumbling Duke)
  - クラス名: (例: sun.applet.AppletViewer)
  - クラスパス: (例: F1:JAVACLS.ZIP;F1:JAVAEXT.ZIP;F1:DUKE.JAR)
  - vmheap容量:
  - Javaメモリー容量:
  - Javaスタック容量:
  - PJava引数:
  - アプリ引数: (例: DUKE.HTM)
  - 環境変数: (例: HOME=F1: JAVA_HOME=F1: USER=ikoma)

JAVAAPL.XMLファイル

```
<?xml version="1.0" encodeing="shift_jis">
<APPLIST>
    <TOTAL>2</TOTAL>
    <IDLIST>0001,0002</IDLIST>
    <ENV>... </ENV>
    <APP>
    <ID>0001</ID>
    <NAME>Tumbling Duke</NAME>
    <TYPE>JAVA_APPLET</TYPE>
    <START>DUKE.HTM</START>
    <CLASSPATH>F1:JAVACLS.ZIP,F1:JAVAEXT.ZIP;F1:DUKE.JAR</CLASSPATH>
    <OPTION></OPTION>
    <NETCONNECT>
        <TYPE>MANUAL</TYPE>
    </NETCONNECT>
    <ENV>
        <ITEM>HOME=F1:</ITEM>
        <ITEM>JAVA_HOME=F1:</ITEM>
        <ITEM>USER=ikoma</ITEM>
    </ENV>
</APP>
<APP>
</APP>
</APPLIST>
```

ZaurusのPersonalJavaで利用できる固有機能

- □Zaurusのソフトウェア・キーボード(手書き文字認識等)
  - ○Touchableのソフトキーボードも選択可能
- □PPPダイアリング
- □カメラ(予定)
- □PIM機能(予定)
  - ○アドレス帳
  - ○スケジュール

インタフェース

- □LANカード、PHS
  - ○PPPダイアリングはAPIを設計
- □フラッシュメモリ・カード
  - ○ファイル・システムとして利用
- □カメラ・カード
  - ○カメラAPIを定義
- □ザウルスのアドレス帳、スケジュール、フォトメモリ、カメラ
  - ○Java APIからアクセス可能
- □ザウルスの入力パッド
  - ○手書き文字認識、ソフトキーボード、等
  - ○Java AWTのテキスト・エリア、テキスト・フィールドへ
- □IR、シリアルはJNI経由で呼び出し可
  - ○別途コーディング要
- □Bluetooth(予定)

現在の制限事項(1)

- □ファイルシステム
  - ○8+3文字のフラット・ファイルシステム
  - ○絶対パスのみ
    - ▹相対パス非サポート
  - ○本体(F0:)とフラッシュメモリカード(F1:)
- □ネットワーク
  - ○マルチキャスト、マルチホスト未対応
  - ○マルチスレッド環境でのDNS参照
  - ○ローカルポートへのループバック
- □RMI
  - ○クライアントのみサポート
- □java.security.*は非サポート
- □SSL、gopher、mailtoプロトコルは非サポート

現在の制限事項(2)

- □AWT
  - ○非サポート
    - ▹ファイルダイアログ
    - ▹メニュー関係
    - ▹スクロールバー
      スクロールボタンで代用
    - ▹モードレスダイアログ
  - ○ダイアログは一時にはひとつのモーダル・ダイアログしか見えない
  - ○フォント
    - ▹1種類のみ。サイズについては近似
  - ○カラー24ビット処理だが描画は16ビット
  - ○カーソル
    - ▹ウィンドウ・カーソルはサポートしない
  - ○サウンド
    - ▹サウンド関連API未サポート

現在の制限事項(3)

- □ネーティブスレッド・スタック
  - ○-ssオプションでCスレッドのスタック領域変更不可(40KB)
- □オートオフ機能
  - ○Javaプログラム実行中はオートオフ機能がOFFになっているので、自動的に電源OFFしない
- □メモリ容量
  - ○-vmheapと-msで指定するサイズで4500KBまで

課題

- □性能
- □メモリ容量
- □Jiniサポート
- □WWWブラウザとの連携
- □J2ME(CDC)対応

メモリが足りない!

- □RAM
  - ○1999年春までのザウルス(MI-610) : 2MB
  - ○1999年春からのザウルス(iCruise, MI-C1) : 8MB
  - ○??? : 16MB(???)
- □最適化のトレードオフ
  - ○メモリと速度を秤にかけりゃ、メモリが大事なPDAの世界
- □基板上のチップのスペースも貴重なリソース
- □メモリを積んでおけば露呈しないバグは多い
- □時間が解決する?
- □アプレットを動かすには
  - ○WWWブラウザとJVMとアプレットの共存
  - ○アプレット動作中にスクロールされたら!?

パフォーマンス

- □参照実装のままではだめ
  - ○Solaris/Windows上の実装
- □チューニングされていない
- □バッテリ駆動
- □メモリと速度のトレードオフ
  - ○JITやHotSpotは使えない
- □最適化
  - ○インタプリタループのアセンブラ化
  - ○ROM化(プリリンク)
- □コンパイラの最適化が効かない!
  - ○Solaris版は速くなるのに!?

ザウルスでのJavaアプリケーションの開発

- □PC/WS上の市販のJava開発環境で開発
  - ○VisualCafe, CodeWarrior等
- □利用できるのはPersonalJava API(1.1.3版)のみ
  - ○javacheckツールで確認
  - ○http://java.sun.com/products/personaljava/より入手可能
- □PC/WS上PersonalJavaエミュレーション環境で動作確認
  - ○http://java.sun.com/products/personaljava/より入手可能
- □作成したJavaプログラム(jarまたはzip形式)をPCからZaurusへ
  - ○フラッシュメモリ・カード
  - ○LAN
- □Zaurus上で動作確認
  - ○(jdb対応未定)

VGA画面のPDAの経験

- □従来、1/4 VGA(320x240)
- □VGA(640x480)にしたら、
- □PCと同一のGUIのまま使える
- □(はずだったが...)
- □表示はできるが物理的サイズが小さすぎて読めない
- □さらにペンタッチはもっとたいへん(List等)
  - ○同じVGAでもPDA用に表示レイアウトの再設計が必要

他社による評価例

- □オムロン/オムロンソフトウェア
  - ○JUMON(モバイルエージェントミドルウェア)
    - ▹(旧ABDM)
    - ▹2000年10月CEATECにてデモ(MI-EX1)
      チャットクライアント
      　　無線LAN使用
      じゃんけんゲーム
    - ▹http://www.jumon-agent.com/
- □KDD研究所
  - ○JaMaPSブラウザ
    - ▹重ね合わせ可能なWebグラフィクスメディア
    - ▹ダウンロードした地図表示アプリケーション
    - ▹2000年6月JavaOneにてデモ(MI-EX1)

Javaの移植方法

- □機種依存部を作成する
  - ○機種独立部分は(原則として)触らない
  - ○Solaris ∩ Windows - ZaurusOS = ?
    - ▹VMで必要な機能
    - ▹AWT等、ライブラリから呼び出されるネイティブメソッド
- □中心となる機能から順次リンクしてテスト
- □Solarisの参照実装と動作比較
- □JCK(Java Compatibility Kit)で互換性の確認

ホストOS上へのJava移植の問題点

- □スレッド
  - ○Natieve Threads or Green Threads?
- □メモリマネジメント
- □グラフィック・ツールキット
  - ○どのレベルで?
- □ファイル・システム
- □コミュニケーション

移植時に必要な機能(1)

- □スレッド
  - ○POSIX準拠のスレッド
  - ○各スレッドのスタック・ポインタを取得できること(GCのため)
  - ○グリーン・スレッドも利用可能
- □64bit整数演算
- □IEEE754準拠の浮動小数点演算
  - ○境界値の処理、NaN、Infの扱いが正しくできること

移植時に必要な機能(2)

- □ファイルシステム
  - ○ディレクトリ、ロング・ファイル名が扱えること
- □ネイティブ関数呼び出し
  - ○JavaのメッソドからC言語関数の呼び出しができること
- □TCP/IPネットワークインタフェース
  - ○socket(),selecct(), read(),write()
  - ○マルチキャスト、マルチホスト

移植時に必要な機能(３)

- □描画関数
  - ○線分、円、楕円、ポリゴン、楕円弧
  - ○塗りつぶし、矩形、楕円弧、ポリゴン
- □イベント入力
  - ○ペンorマウス
- □日本語処理
  - ○入力メソッド、フォント

ROM化

- □組込み機器ではROMよりRAMが貴重
- □起動時間の短縮
  - ○事前にクラス・ローディングし、ROMイメージへ
- □CodeCompactツール

参照実装の実際

- □そのままでは商品にならない
  - ○高速PC上でちゃんと動いただけではだめ
- □コンシューマ市場をターゲットにするなら、参照実装の実行環境ももっと
  - ▹ (Sunへのお願い)
  - ○遅い (安い、低消費電力) プロセサ
  - ○少ないメモリ

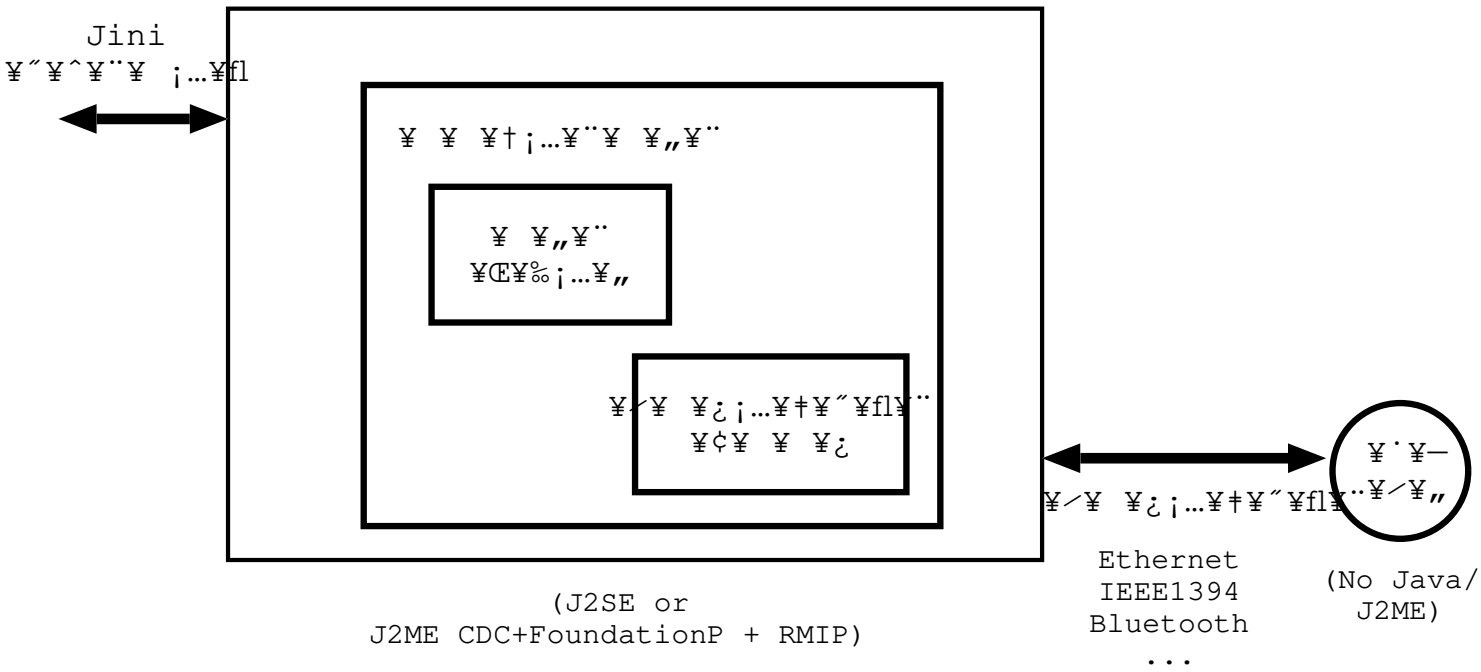
Jiniの実装

- Jiniのコードは大きくない
  - Javaプラットフォームは別
- Java VMが前提
  - 現状Java2のみ
  - PersonalJava現行版では未サポート
  - Java2 MicroEdition (J2ME)CDC/FoundationP/RMIP
  - ただし、PC等でプロキシも可能
    - サロゲート・アーキテクチャ
- プロトコル
  - ディスカバリを除き、すべてRMI
    - 現状のRMIは+TCP/IP|上に実装
  - ディスカバリにはTCP/IP(UDPマルチキャスト)を使用
  - 現状のJiniはTCP/IP前提だが、原理的には他プロトコルでも可
    - Bluetooth

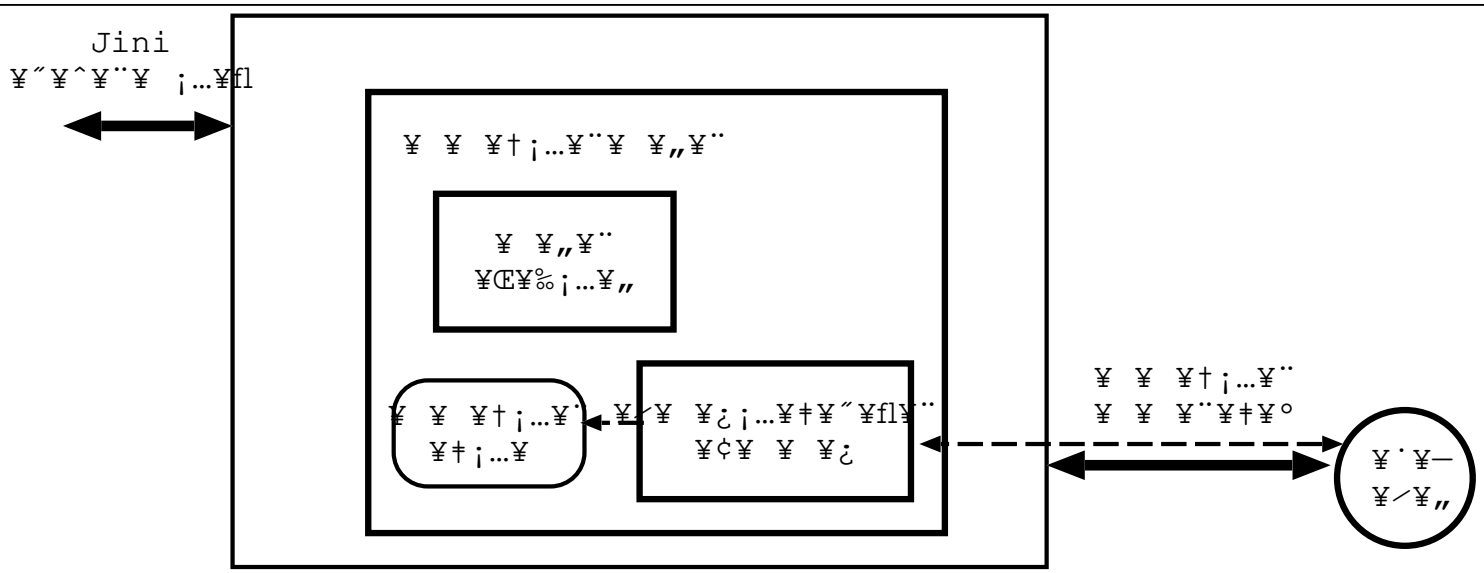
Jini Surrogate Architecture

- リソースに制限のあるデバイスをJiniサービス/クライアントに
- サロゲート・ホストにコードをアップロード
  - サロゲート・プロトコル
- サロゲート・ホスト環境下でコードを実行
  - プライベート・プロトコルでデバイスと通信
- サロゲート・ホストはデバイスの状態を監視
- インターコネクト
  - Jini技術と物理的ネットワーク技術を分離
- Discovery
  - Look Up Serverでなく、適切なSurrogate Hostをみつける

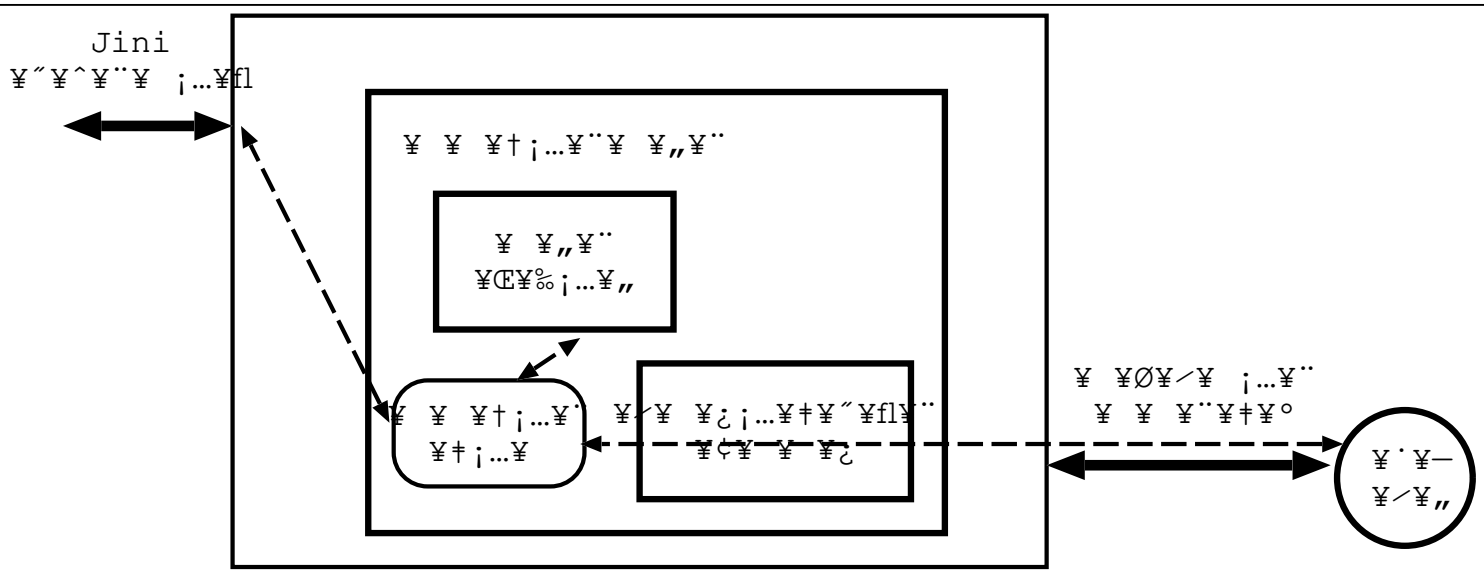
Jini Surrogate Architecture



Jini Surrogate Architecture



Jini Surrogate Architecture



JiniとPDA

- □現状のPersonalJava(1.2)ではJiniは未サポート
- □Jiniクライアントとしての PDA
  - ○汎用リモコン
  - ○ネットワーク・サービス・ブラウザ
  - ○プリンタの利用
- □Jiniサービスとしての PDA
  - ○PIMサービス
  - ○ディジタル・カメラ・サービス
  - ○個人情報サービス
    - ▹購入履歴、治療・投薬履歴、洋服サイズ、...
- □無線技術とJini技術の組合せ
  - ○Bluetooth + Jini

JavaOne2000でのJini共同デモ

- セイコーエプソン、サン・マイクロシステムズ、シャープ
- コンセプト:
  - 「日本メーカーが実際のデバイスをJiniで動かして見せる」
    - プリンタ、カメラ、PDA、...
    - PCは使わない(見せない)方針
    - ワイアレス接続

NAI Forum

- Network Application Interface Forum
- Java/Jini技術の広域ネットワークへの適用
- 1大学、13社
- http://www.nai-forum.org/

さいごに

PDAでJavaをどう使いますか？