6.　エージェント型ソフトウェアのモビリティに関する仕様策定と実装

6.1.　はじめに

本研究は、PDA や JVM 搭載携帯端末などモビリティを有するデバイスにおいて、OpenSOAP のプラットフォームを利用して Web サービス等のネットワークアプリケーションに対応させることで、OpenSOAP コミュニティとしてのウェアラブルな利用形態を提案することを目的としている。このために、JVM 搭載携帯端末やモバイル向け Java アプリケーション開発環境の最新動向を調査し、これに基づいて、現状に即したモバイル型 SOAP クライアントおよびサンプルアプリケーションの仕様策定を行った。

上記調査結果に基づき、種々の問題点を残しながらも現状において最大の市場規模と最高の開発環境を提供する i-αppli プラットフォームを開発フィールドとして選択し、これに適用するための詳細な仕様を策定するとともに、実装を行った。ここでは、携帯端末に搭載するアプリケーションの規模の小ささから、これを支援する Web サーバとの連携型のクライアントシステムを構築した。サンプルアプリケーションとして、OpenSOAP コアパッケージが同梱提供する Calc サービスに対する携帯端末上クライアントを実装した。

さらに、エージェント型ソフトウェアによるネットワークサービスの知的自律化を目的として、SOAP メッセージの自動応答や自律的判断機能を搭載したアプリケーションの開発と評価を行った。従来の携帯通信アプリケーションは、ネットワークの末端部を身に付けることで、時と場所を選ばない、ユーザからの能動的なネットワークアクセス支援を行ってきた。これに対し、あらかじめ携帯端末にプログラマブルなアクセススケジュールを入力し、さらに入手した情報に対する応答・処理戦略を端末が内蔵できれば、ユーザは情報が処理された結果を常に携帯し、これを確認することができる。複雑な処理が要求される場合は、プログラムされたルーチンを解除し、ユーザが自ら応答することもできる。

このような知的エージェント型ネットワークモバイルアプリケーションを目指し、本研究では、OpenSOAP の枠組みと連携して稼動する Java アプリケーションを設計し、実装を行った。この実現のために、OpenSOAP コアスペックへの機能の提案なども一部行い、プロジェクトのポテンシャルの拡充をも図っている。

以上のように、OpenSOAP のフレームワークをモバイルプラットフォームに適用することにより、Web サービスを視野におき、SOAP を実装することで情報を汎用的に運用可能な、即時的で高度なサービスの形態を提案・実証した。

6.2. ソフトウェアアプリケーションのモビリティに関する現状

6.2.1. モバイル向け SOAP 実装の現状

まず、世界におけるオープンソースモバイルネットワークの実状を把握し、本研究の位置付けを明確にするために、以下に本調査実施時点（２００１年１０月１日現在）での、公開済みモバイル向け SOAP 実装を列挙する。

- **kSOAP** (Enhydra.org)
  概要：J2ME（6.6 用語集参照）の SOAP API、kMXL(同プロジェクト)に基づいている。
  バージョン：0.95（2001/9/25）
  http://ksoap.enhydra.org/
- **pocketSOAP** (Enhydra.org)
  概要：WinCE（pocketPC）用 SOAP クライアント。COM コンポーネントとして実装されている。
  バージョン：1.1beta4(2001/9/19)
  http://www.pocketsoap.com/
- **The Bubbles Project** (Vivek Chopra)
  概要：KVM で動作する Java クライアント
  バージョン：未発表(2001/10/1)
  http://www.soaprpc.com/software/bubbles/
- **HORB extension SOAP** (HORB Open)
  HORB（6.6 用語集参照）の SOAP 対応。i-αppli でも動作が確認されている。
  バージョン：未発表(2001/10/1)
  http://soapc.sourceforge.net/

これらからわかるように、当該分野は未だ未成熟であると同時に、急速に展開を続けている。同時に、現存するモバイル向け SOAP 実装はそのほとんどが Java による実装であり、OpenSOAP プロジェクトの独自性を優位性を充分に主張できる領域である。すなわち、新規標準技術となりうる Java フレームワークと、レガシシステムの中核である C 言語との融合を、モバイルという利用形態のひとつの新しい標準として地位を獲得しつつあるプラットフォーム上で実現することで、急速に発展するインターネットテクノロジーに貢献するものである。

次節では、これらアプリケーションのプラットフォームとなる、日本における Java 搭載携帯端末の実状を調査した結果を述べる。

6.2.2.　モバイルプラットフォームの現状調査

- i-αppli（NTT DoCoMo）

i-αppli（以下、i アプリ）とは、2001 年 1 月に開始された、NTT DoCoMo の「i モード」対応携帯電話で利用できるアプリケーションサービスのことである。「503i」シリーズ以降の対応した携帯電話には、Java 言語で作成したプログラムの実行環境（Java 仮想マシン：JVM）が搭載されており、Java 言語で作成したアプリケーションソフトをダウンロードして実行することができる。従来の i モードサービスでは実現できなかった、アクションゲームや随時更新される株価チャート、スクロールできる地図などの「動きのある」コンテンツの提供が可能となり、携帯電話の利用方法の幅を大きく広げるものとして期待されている。ただし、各端末に独自の機能（API）があることによる非互換性や、通信速度や内蔵メモリ容量などとの兼ね合いによる厳しいサイズ制限など、限界や問題も指摘されている。以下に詳細に述べる。

i アプリは、プラットフォームが携帯電話という小規模性から、1 本あたり 10K バイトまでという容量制限があるため、中規模なアプリケーションソフトさえも稼動することは不可能である。また、1 台の端末で同時に複数の i アプリを起動したり、ほかの i アプリと連動させるような使い方も許容されていない。また、i アプリが端末内でアクセスできるメモリ領域（Scratch Pad）は個々に指定されており、他の i アプリのデータや本体の電話帳データを参照したり、書き換えたりという動作も許可されていない。しかしながらそれゆえに、Java を利用した携帯ウィルスによる被害はまず発生しないという、セキュリティ面での安全性を確保している。

また、i アプリでは、テキストのほかに画像などを扱うことができるが、これまでの i モードコンテンツのような GIF ファイルの差し替えではなく、端末上で元データから描画させることができるため、結果として伝送するデータ量を少なくできるというメリットがある。例えば、始めに天気図で使う日本地図を画像ファイルとして送り、刻々変わる気象情報はデータから地図上に描き起こすことができる。

このような枠組みを、ある一定間隔で自動的にサーバへアクセスするように構築すれば、常に最新のデータを読み出し、気象情報や株価情報のチェックなどへの応用が考えられる。

こうしたネットワーク自動アクセスの機能を NTT DoCoMo はエージェント機能と呼んでいる。エージェント機能を使うことで、ユーザにとっては擬似的に常時接続されたパソコンと同様の環境となり、商品在庫やグループメンバーのスケジュールなどが最新情報として表示できるなど、ビジネスユーティリティとしても魅力的な i アプリが期待される。

503i シリーズは、Java 2 Platform Micro Edition (J2ME)の Connected Limited Device Configuration (CLDC) (6.6 用語集参照) がサポートされている。この「J2ME/CLDC」は、PDA などの携帯端末向けに設計された「K Virtual Machine (KVM) (6.6 用語集参照)」をベースに作成されており、携帯電話などの小型デバイスで使うための最適な Java プラットフォームとなっている。i アプリを実行するには、J2ME/CLDC に加えて、独自に用意された i モード拡張ライブラリの構成が必要であり、端末メーカーの提供する機種ごとの拡張ライブラリも利用可能である。以下に i アプリのアーキテクチャを図示する。(図 6.2.1)

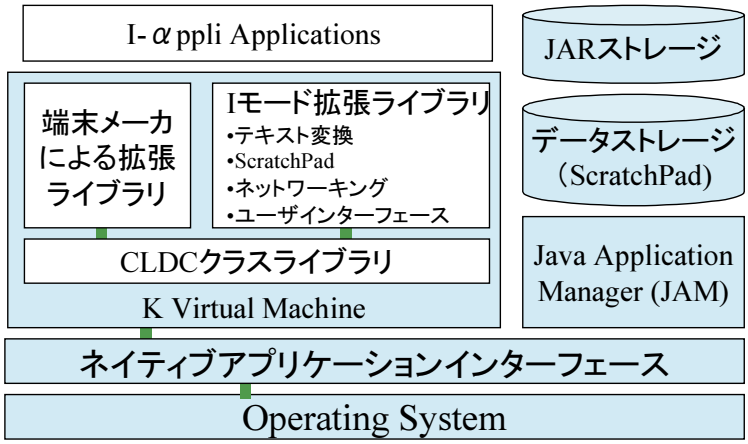

図 6.2.1 i モード対応 Java 搭載端末アーキテクチャ

i アプリの Java 開発環境としては、NTT DoCoMo が提供する、J2ME Wireless SDK for the DoJa release2.2 (http://www.nttdocomo.co.jp/p_s/imode/) や、ZENTEK の i-JADE Lite for x503i version1.3.0 (http://www.zentek.com/i-JADE/) が利用可能である。J2ME Wireless SDK for the DoJa は、基本エミュレータ、クラスライブラリ、GUI 開発環境、各種ユーティリティを含み、統合開発環境・Forte for Java との統合が可能である。また、ZENTEK の i-JADE Lite for x503i は、各メーカ毎に異なる詳細仕様を再現するための個々のエミュレータを用意しており、各種開発環境に統合可能となっている。これら開発環境はともに無償でダウンロード可能である。

- ezplus（au/KDDI）

  KDDI の ezplus の主な特徴としては以下のようなことが挙げられる。
  - プログラムサイズは 50K バイトまで許容。
  - プラットフォームとして KVM/MIDP、アプリックスの JBlend を搭載。
  - Java アプリケーション同士の通信方法（P2P）を装備。
  - セキュリティレベルに応じて端末内のデータの利用が可能。

先行する NTT DoCoMo の i アプリと異なり、KDDI では KVM/MIDP という世界標準の仕様を採用する。さらに、JavaVM にはアプリックスの JBlend を全機種に採用し、互換性を高めているだけでなく、どの端末にも Qualcomm 製のチップが採用されており、i アプリで指摘された実行環境の差を解消し、仕様から Java の実行速度まで統一された実行環境となる。

大きな特徴は、Java プログラムである KJX（Kddi Java eXtentions）の最大サイズが 50K バイトであることである。携帯電話キャリア各社の Java 環境と比較しても、この時点での KDDI の Java プログラムサイズは最大となっている。

ezplus では、HTTP 通信がサポートされていない代わりに、Java アプリケーション間の通信方法が用意されている。すなわち、Peer to Peer による Java アプリケーション同士の通信が可能になっている。これを活用するために、オムロンが開発した P2P サービス実現のためのミドルウェア、「Jumon」が搭載されている。他携帯の Java プログラムを操作する「ORB」（Object Request Broker）機能や、携帯間でエージェントを移動させて、相手携帯で処理を行った後、結果を返信するといったモバイルエージェント機能も可能となっている。通信方式としては au の保有技術である新 C メール方式を利用する。

さらに i アプリでは利用を許可されなかった、端末の内部情報や個人情報も、一部利用可能とした。

ezplus も、i アプリと同様に、KVM（K Virtual Machine）の上に CLDC が実装されている。i アプリとの違いは、GUI などのライブラリとして MIDP（Mobile Information Device Profile）を全面的に採用している点である。すなわち、i アプリでは、GUI などの実装に関して独自のプロファイルを採用している一方、ezplus で採用した MIDP は Java 標準として策定されたプロファイルの 1 つで、携帯電話や PDA などで汎用的に設計できるようになっているのが特徴である。

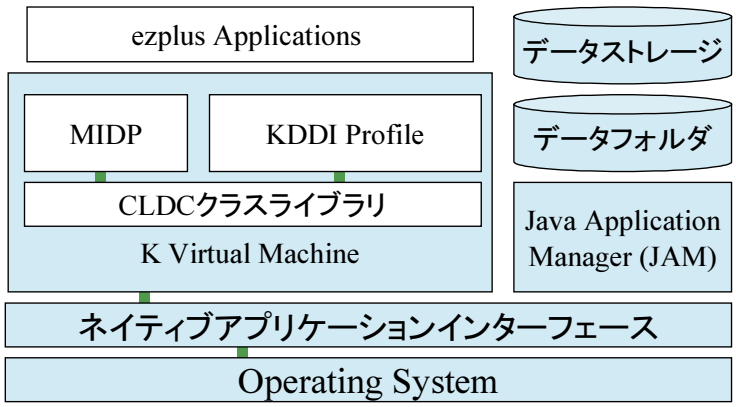

図 6.2.2 ezplus Java 搭載端末アーキテクチャ

ezplus 関連の開発ツールとドキュメントは以下のように入手可能である。ezplus のアプリケーションの作成には、まず Java2 の開発環境が必要である。標準の JDK が http://wwww.javasoft.com から入手できる。その上で、「Java2 Platform Micro Edition」「Wireless Toolkit」が必要である。ツールキットには、Windows 上で動作する KVM、i アプリ開発でも使用される簡易開発環境 KToolbar、CLDC や MDIP のライブラリセットが含まれている。ツールキットは
http://developer.java.sun.com/developer/earlyAccess/j2mewtoolkit/
からダウンロードできるが、事前にユーザ登録が必要となる。

さらに KDDI 独自のプロファイルに実装されているライブラリなどが収録された作成ツールも必要になる。KDDI が配布しており、
http://info.ezweb.ne.jp/factory/tec/spec/ezplus3.html からダウンロードできる。

関連ドキュメントとしては、まず KDDI 自身が配布している「プログラミングガイド」「特殊文字コードセット一覧」「API マニュアル」（KDDI 独自のプロファイル分のみ収録）の 3 点が
http://info.ezweb.ne.jp/factory/tec/spec/ezplus.html から入手できる。

さらに、開発には MIDP のドキュメントが必須である。
（http://java.sun.com/products/midp/）

- J-Sky Java アプリ（J-Phone）

J-Phone の Java 搭載端末の特徴は以下のとおりである。

- J2ME/MIDP に準拠。
- ２D スプライトや３D ポリゴンなどの強力な API を実装。
- 30K バイト以上のダウンロードサイズ。
- 将来的に API をオープン化。

NTT DoCoMo の Java 搭載携帯電話「503i」シリーズが、半ば独自仕様なのに対して、J-Phone は au と同様、MIDP と呼ばれる携帯機器向けの標準的な仕様を採用する。Java アーカイブのサイズは 30K バイト前後であり、503i シリーズでは、サイズが 10K バイトに制限されていることから、開発者からは制限の緩和を求める声が多く、J-Phone に優位性がある。JavaVM には、アプリックスの「J-Blend」を採用する。DoCoMo の端末が様々なメーカの JavaVM を採用しているのに対し、J-Phone 端末では J-Blend に統一する。端末メーカ間で共通の JavaVM を使うことで機種間の互換性が高いというメリットがある。

MIDP は標準的な API だが、マルチメディア機能で、MIDP は競争力上問題があるとされる。そのため J-フォンがアプリックスと共同開発したのが、“ケータイでは類を見ない拡張 API である JSCL（J-PHONE Specific Class Libraries）である。（図

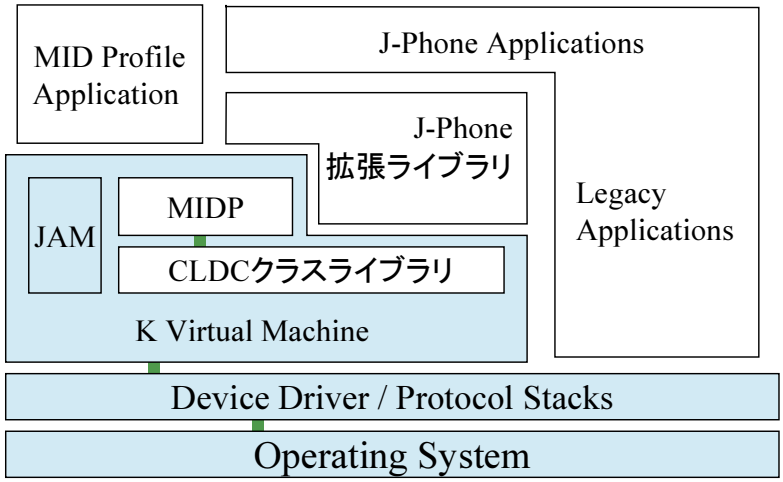

図 6.2.3 J-Phone Java 搭載端末アーキテクチャ

6.2.3）

この拡張によって、2D のスプライト処理や、3D ポリゴンエンジンも Java から利用可能になり、携帯電話の表現力を大幅に拡張する仕様となっている。

また、電話機能やメール機能の制御が Java アプリケーションから行えるのも特徴であり、大幅な機能拡張が行われる J-フォンの Java 仕様だが、その分、セキュリティが犠牲になっていることは否めない。2001 年 6 月のサービス開始時は、J-フォンが検証し、J-フォンが保有するダウンロードサーバ内にある Java アプリケーションしかダウンロードは行えなかった。

NTT DoCoMo の 503i シリーズの場合、ハードウェアのアクセスなどセキュリティ上の問題となる機能はすべて不可とした代わりに、サービス開始当初から一般サイトでも Java アプリケーションが提供できた。J-フォンの Java は、多くの拡張を施し、ハードウェアの制御も部分的に認めたことで、アプリケーションの自由度は上がったものの、当初は“オープン”とは呼べないものとなっている。

J-フォンの拡張 API の内訳は以下のようになる。

１．デバイスコントロール

携帯電話のハードウェア情報を取得したり、制御したりする拡張 API。電界強度や、「ステーション」などで実用化している位置情報、バッテリー残量などをアプリケーションが取得できる。また、LCD バックライトや LED、バイブレータの制御も行える。

2．ウィンドウマネージャー

MIDP の低レベルグラフィック機能を拡張するもので、画面の縦分割を可能にし、タイトル部とゲーム、映像部分を個別に表示できる。

3．日本語・絵文字対応

MIDP の高レベルグラフィック機能の拡張で、文字入力時の日本語対応、GUI メニュー部品の日本語表記、GUI メニュー部品に動く絵文字ボタンなどを追加できる。

4．メディアプレーヤー

端末内に登録されたメディアデータの再生が行える。対応するフォーマットは JPEG アニメーション、PNG アニメーション、SMAF 形式の楽曲、SMD 形式の着信メロディとなる。

5．サウンド

複数のフレーズ（メロディ）を独立して再生できる。ゲーム時の BGM と効果音などに利用できる。音にイベントを埋め込み、動画や映像との同期も可能。同期機能はヤマハ製のサウンドチップ MA2、MA3 をサポートする。

6．2D スプライト

セガ／スマイルビット社と共同開発した 2D スプライトエンジンを搭載する。1 スプライトデータは、各画素で 256 色、8×8 ピクセルのデータで構成され、スプライト単位で回転、反転、色指定、透過処理が可能。最大 256 個のスプライトデータ配列の定義が行える。秒間 8 フレーム以上の動画性能を持つという。また、2D のスプライト・フレームバッファもサポートし、背景とスプライトデータの合成に利用できる。さらに 2D スプライト用の演算機能として、三角関数などを含む固定小数点演算、2D ベクトル演算機能も備える。

7．3D ポリゴン

バンダイネットワークスとエイチアイが開発した 3D ポリゴンエンジンも搭載し、制御用の拡張 API が追加される。

8．テレフォニー

Java アプリケーションから、電話の発着信を制御できる機能。電話の着信が Java アプリケーションに通知される機能と、アプリケーションから電話を発信できる機能がある。

9．メール通信

Java アプリケーションから、メールの送受信を制御できる機能。メールの受信がアプリケーションに通知される機能と、受信メールの各種情報、新規メールをアプリケーションから送信できる機能がある。

J-Phone は 2002 年 3 月、携帯電話向け Java サービス「Java アプリ」を一般に開放した。これまで Java アプリを提供できるのは公式コンテンツプロバイダに限られていたが、一般のユーザが作成した Java アプリを動作させられるようになる。

開発ツールも無償で提供され、既に発売されているパケット対応端末からダウ

ンロード可能になる。パケット対応端末では Java アプリサイズが 100K バイトに拡大され、機能拡張も施されている。

“一般に公開”といっても、セキュリティ確保のために、どのサイトからでも Java アプリをダウンロードできるわけではなく、J-Phone が承認したサイト、つまりコンテンツアグリゲータのサイトからのみのダウンロード可能となる。Java アプリ作者はコンテンツアグリゲータに開発者登録し、作成した Java アプリのチェックが終わるとコンテンツアグリゲータのサイトに登録されるという流れになる。つまり、多彩な機能を持つ Java アプリだけに、不正なアプリが作られないよう、作者の身元確認や作成されたアプリのチェックが行われる。配布元としてサーバの管理も行い、アプリに問題が出た場合、すぐに配信をストップできるようになっている。

J-Phone では、「J-PHONE Developer Program」（http://www.dp.j-phone.com/）にて、無料で技術情報を提供。法人向けにテクニカルサポート付きの有償サービス「J-PHONE Certified Solution Provider（JCSP）」も提供する。

6.2.3.　最新動向

J-Phone だけでなくモバイル環境において高機能ネットワークサービスを享受する仕組みは 2002 年になってからも急速に進化している。いわゆる携帯端末における第二世代 Java アプリケーションと呼ばれる枠組みは、これまでの各社の欠点を補い、Web サービスを指向するにあたりより現実的な仕様を提示している。図 6.2.4 に各社の対応をまとめたものを図示する。

	主な変更点	開始時期
DoCoMo	**アプリサイズが10Kバイトから30Kバイトに。** スクラッチパッドの大幅な拡張。通信速度が9.6Kbpsから28.8Kbpsに高速化	2002年春
J-Phone	アプリサイズが100Kバイト（レコードストア含む）に。通信速度が9.6Kbpsから28.8Kbpsに高速化。**一般の開発者が公開可能に**	2002年3月
au	**HTTP通信およびHTTS通信が可能に。** gpsOneなど位置情報機能も利用可能に	2001年12月

図 6.2.4 第二世代 Java アプリケーションへの移行状況

6.3. OpenSOAP MP クライアントの仕様策定と実装

上記調査結果から、仕様策定・実装作業開始段階において、容量やセキュリティ上のネットワーク接続制限など、種々の問題点を残しながらも現状において最大の市場規模と最高の開発環境を提供する i-αppli プラットフォームを開発フィールドとして選択した。本節では、これに適用するための詳細な仕様を策定するとともに、サンプルクライアントの実装を行ったのでこれについて説明する。ここでは、携帯端末に搭載するアプリケーションの規模の小ささから、これを支援する Web サーバとの連携型のクライアントシステムを構築した。すなわち、接続先を i-αppli ダウンロードサーバのみに限定されているため、このサーバを SOAP メッセージ作成や、記憶装置として携帯端末の能力を支援する位置付けとし、携帯端末とサーバを対としてクライアントとみなすものとする。本フレームワークを OpenSOAP MP（Mobile Phone）と呼ぶ。携帯端末、i-αppli ダウンロードサーバそれぞれの役割と機能を図 6.3.1 に示す。携帯端末上 JVM には、入力や結果出力それぞれの画面設計とパラメータ入力部のサービスインターフェース部と、i-αppli の特性を活用したエージェント機能部が搭載され、通信を行う i-αppli ダウンロー

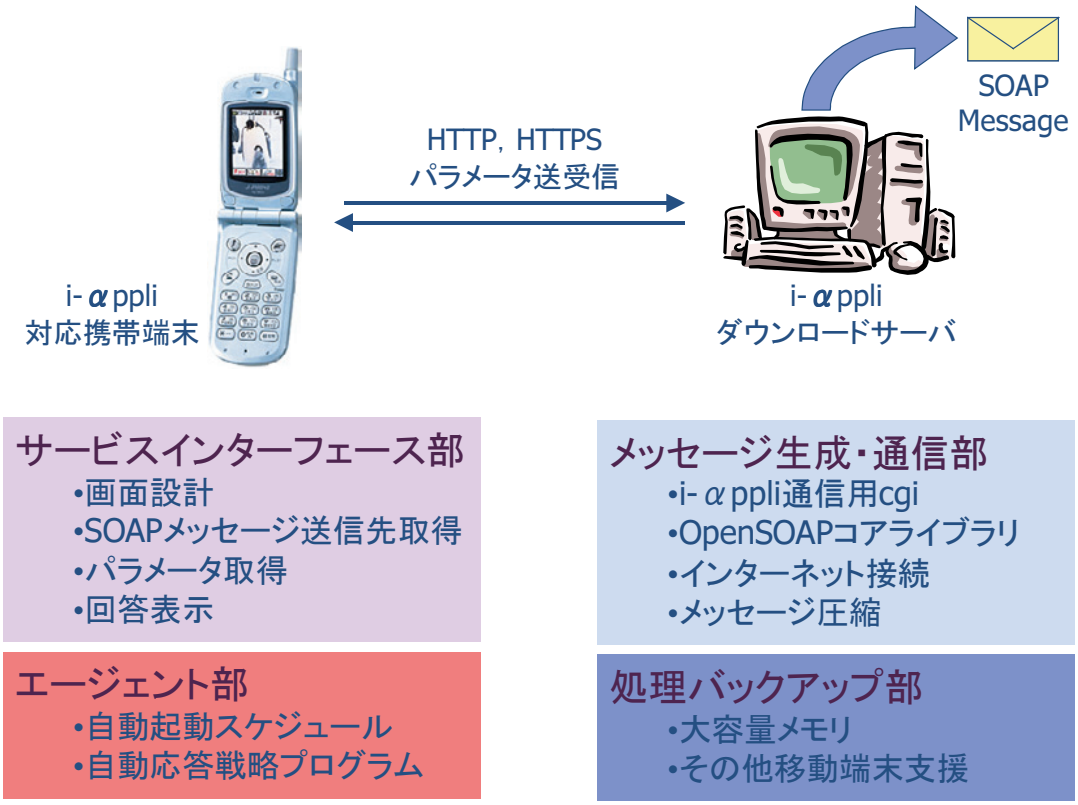

図 6.3.1 OpenSOAP MP 仕様概要

ドサーバには、i-αppli とパラメータの送受信を行い、それをもとに目的とする SOAP メッセージを生成、インターネットを通じて Web サービスに送信し、受け取ったレスポンスメッセージを解釈する OpenSOAP API を用いたフルスペックの SOAP クライアントプログラムを稼動させる。さらに i-αppli ダウンロードサーバには、クライアントプログラムを支援する大容量記憶装置や、その他支援プログラムを搭載することができる。

この仕様を実証するため、ここではサンプルアプリケーションとして、OpenSOAP コアパッケージが同梱提供する Calc サービスに対する携帯端末上クライアントを実装した（図 6.3.2）。Java で動作するエミュレータ上の画面にはパラメータを入力するフィールド等が表示され、携帯端末のボタンから入力することが出来る。エミュレータ上の端末の「送信」に割り当てられたソフトキーを押下すると、以下のようなパラメータ文字列が生成され、i-αppli ダウンロードサーバに HTTP プロトコルをもって送信される。

パラメータ文字列の例：end_point=http://opensoap.jp/cgi-bin/CalcService.cgi&
operation=0&parameter1=52.125&parameter2=25.48

i-αppli ダウンロードサーバはこの文字列をパラメータ要素に分解し、SOAP メッセージを生成、OpenSOAP サーバに管理された Web サービスに送信する（図

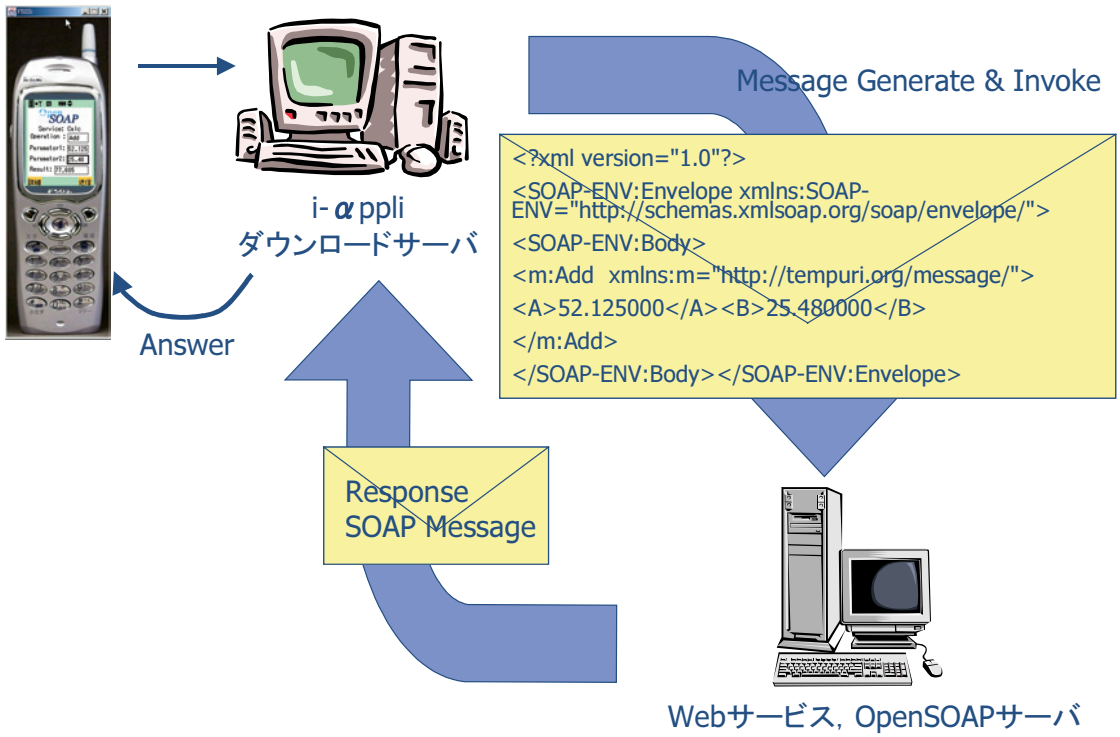

図 6.3.2 OpenSOAP MP サンプルアプリケーション

6.3.2)。OpenSOAP API フルパッケージが導入された i-αppli ダウンロードサーバは返答メッセージをパースし、結果を再びパラメータ文字列を生成して携帯端末に送信する。文字列を受信した携帯端末（エミュレータ）は結果画面を表示する。本サンプルプログラムは、OpenSOAP プロジェクトの WWW サーバから無償でダウンロード可能である。（http://www.opensoap.jp/）

6.4. OpenSOAP MP エージェント型アプリケーション

ここで提案するエージェント型アプリケーションは、前節の OpenSOAP MP 仕様に準拠し、OpenSOAP のフレームワークを汎用的に有効活用する仕組みを提供するものである。ここで想定する仕組みは、OpenSOAP サーバのオリジナル機能である、非同期処理に関するユーティリティアプリケーションである。すなわち、非同期処理を要するサービスリクエストの処理終了通知の携帯端末での受け取り、もしくは、処理完了の問い合わせを携帯端末から行うというものである（図 6.4.1）。

まず、OpenSOAP MP クライアントは、非同期サービスリクエストを行った OpenSOAP クライアントと通信し、対応するリクエストのメッセージ ID を受け取る。i-αppli はエージェント機能を使ってユーザの指定したスケジュールに従い、定期的に OpenSOAP サーバに処理の完了を問い合わせる。この際、現行の OpenSOAP の使用では、完了している処理に対して問い合わせを受信した場合、対

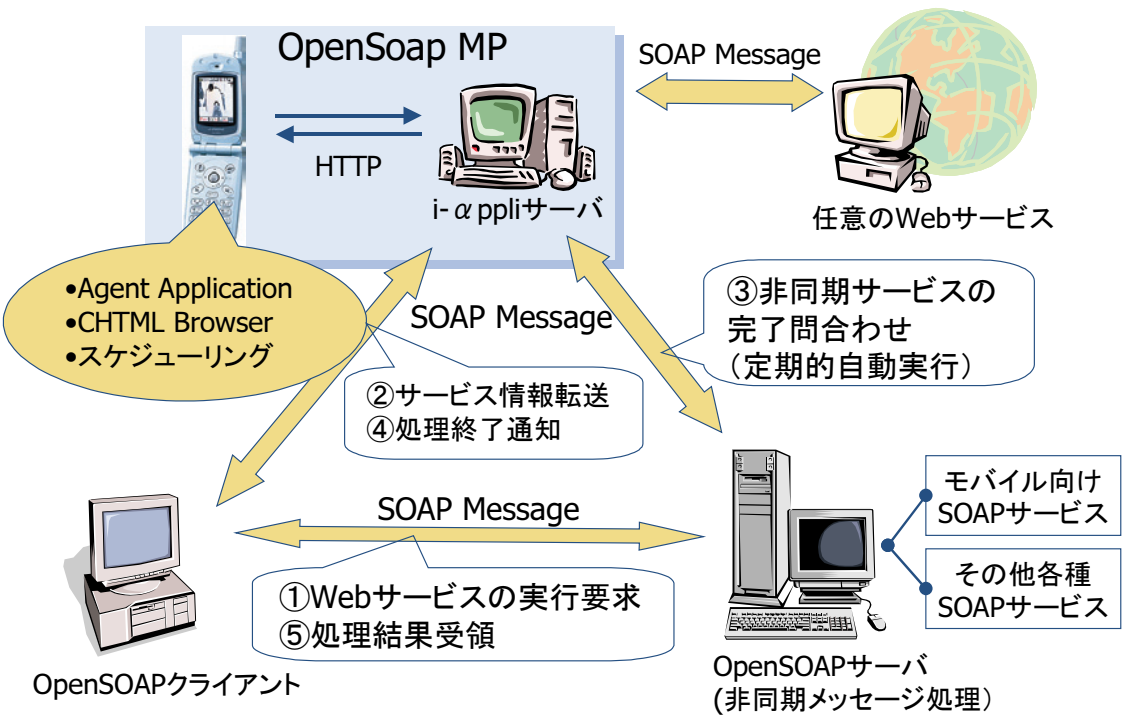

図 6.4.1 OpenSOAP MP エージェント型ネットワーク

応するレスポンスメッセージを、問い合わせをしたクライアントに返信し、スプールから返信内容を削除するという処理を行うが、i-αppli ではレスポンスメッセージの内容を処理しきれない。

このため、ここでは問い合わせのみを行い、処理が終了していたとしても OpenSOAP スプールに内容を保持する仕組みが必要となるため、OpenSOAP サーバの仕様拡張を提案した。図 6.4.2 は拡張された機能を稼動させるためのメッセージヘッダである。これにより、OpenSOAP MP クライアントは、処理終了通知（レスポンスメッセージを受信することも可能なので、簡易的に携帯端末に結果表示を行うこともできる。）を受信し、リクエストを行った OpenSOAP クライアントに処理の終了を通知することができる。

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/
        encodingStyle="http://schemas.xml.soap.org/soap/encoding/">
<SOAP-ENV:Header>
<opensoap-header:opensoap-header-block
xmlns:opensoap-header="http://header.opensoap.jp/1.0/">
    <opensoap-header:message_id>
        TransactionControl.opensoap.jp.2002022021010200001
    </opensoap-header:message_id>
    <opensoap-header:message_undelete>True</opensoap-header:message_undelete>
</opensoap-header:opensoap-header-block>
</SOAP-ENV:Header>
<SOAP-ENV:Body />
</SOAP-ENV:Envelope>
```

図 6.4.2 OpenSOAP MP エージェント型アプリケーションのためのメッセージヘッダ拡張仕様

インターネット技術において、プッシュ式に処理結果をクライアントに受信させるには、別の枠組みが必要となるため、ウェアラブルに結果を入手できるプラットフォームと合わせて、Web サービスの分野において本アプリケーションが有効に作用するものと考える。

6.5.　まとめ

本研究においては、PDA や JVM 搭載携帯端末などモビリティを有するデバイスにおいて、OpenSOAP コミュニティとしてのウェアラブルな利用形態を提案した。このために、JVM 搭載携帯端末やモバイル向け Java アプリケーション開発環境の最新動向を調査し、これに基づいて、現状に即したモバイル型 SOAP クライアントおよびサンプルアプリケーションの仕様策定を行った。

上記調査結果に基づき、i-αppli プラットフォームを開発フィールドとして選択

し、Web サーバとの連携型のクライアントシステムを構築した。サンプルアプリケーションとして、Calc サービスに対する携帯端末上クライアントを実装した。

さらに、エージェント型ソフトウェアによるネットワークサービスの知的自律化を目的として、OpenSOAP の枠組みと連携して稼動する Java アプリケーションを設計し、実装を行った。

本研究の進行にあたり、当初、図 6.5.1 に示す年間スケジュールを計画し、ほぼ予定通りの進捗と成果を得た。

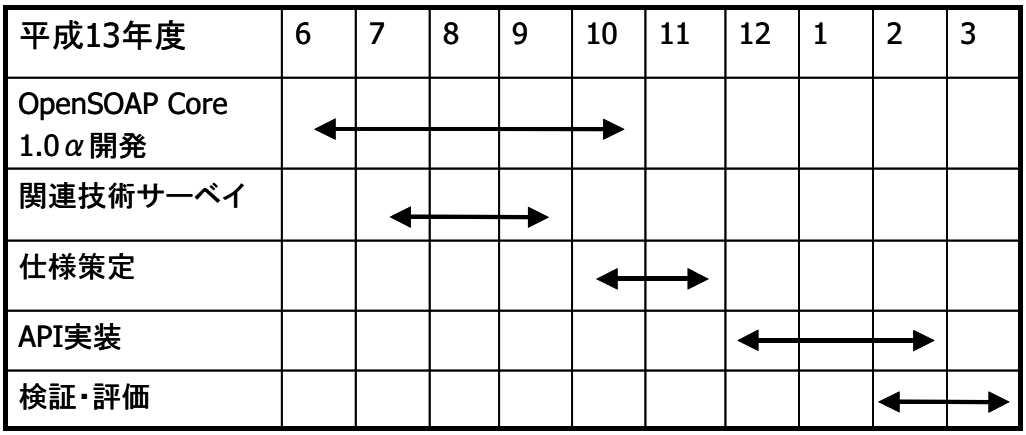

図 6.5.1 本サブテーマの研究スケジュール

6.6. 用語集

- J2ME（Java 2 Micro Edition）

Sun Microsystems 社のプログラミング言語「Java 2」の機能セットの一つで、家電製品や携帯情報端末、携帯電話などの組み込み機器向けの機能をまとめたもの。「CDC」(Connected Device Configuration)と「CLDC」(Connected Limited Device Configuration)という 2 つの想定環境(コンフィギュレーション)に分かれており、前者はカーナビや高性能 PDA といった 32 ビット CPU と十分なメモリを持った環境を、後者は携帯電話やネットワーク家電、通常の PDA などの、低速な CPU と少ないメモリからなる環境を対象としている。現在のところ、一般に J2ME として知られているのは CLDC であり、KVM が仮想マシンとして採用されているのもこちらになる。サーバ向けの J2EE、パソコン向けの J2SE との大きな違いとして、基本部分には仮想マシン(実行環境)と最小限のコア API(中核機能)だけを持ち、デバイス(機器)の種類ごとに定義された「プロファイル」と呼ばれる API やクラスライブラリをそれに付加していくことで機能を

補うという方式を採用している。プロファイルの例としては、携帯電話や通信機能を持った携帯情報端末向けに定められた CLDC 用の MIDP などがあり、これらは Sun Community Process を通して業界ごとに定義されている。

- HORB（Hirano Object Request Broker）

通産省電子総合研究所の平野聡氏によって開発された、分散オブジェクト技術(異なるマシン間でオブジェクト同士がメッセージをやりとりするための技術)の一つ。

- MIDP（Mobile Information Device Profile）

J2ME/CLDC 用のプロファイルの一つで、携帯電話などの携帯端末向けに定義された Java 実行環境の仕様。携帯端末用のユーザ・インタフェースやクラスライブラリなどの情報を含んでいる。携帯電話などで Java を使うためには、J2ME/CLDC 自体に加えて、このプロファイルが定義するライブラリなどを用意する必要がある。NTT ドコモの i モード用の Java 実行環境も CLDC に準拠したプロファイルの一つだが、MIDP とは互換性はない。

- CLDC（Connected Limited Device Configuration）

組み込み機器向けの Java 言語仕様である「J2ME」の一部として定義されている想定実行環境(コンフィグレーション)の一つで、携帯電話や PDA などの小型端末を対象としたもの。カーナビなどの大型端末向けの CDC に対して、CPU 速度もメモリ容量も限られた小型端末を対象としているのが CLDC である。仮想マシンには KVM が採用されており、数百キロバイトのメモリと 16 ビット CPU でも動作させることができる。CLDC に準拠した代表的なプロファイルには、MIDP や NTT ドコモの i モード用 Java がある。

- KVM（K Virtual Machine）

Sun Microsystems 社の開発した、組み込み機器向けの Java 仮想マシン。「K」の名の由来は、必要なメモリがキロバイト(Kilo Bytes)単位で済むことにある。J2ME の核となっているコンポーネントで、ごく小さい容量と 128K 程度のメモリで Java を動作させることを目的として開発された。携帯電話やポケットベル、携帯情報端末、セットトップボックス、POS 端末などでの使用を想定している。およそ 40KB という極めて小さなサイズを特長とするが、他の Java 仮想マシンに比べて Java アプレットやリアルタイム機能が実装されていないなど、機能面では幾分劣っている。