

M-16

## 携帯端末のための FIPA 準拠小型 エージェントプラットフォームの実装

### A Light-weight and FIPA-compliant Agent Platform for Mobile Devices

西山 智† 服部 元‡  
Satoshi Nishiyama Gen Hattori

小野 智弘† 堀内 浩規‡  
Chihiro Ono Hiroki Horiuchi

#### 1. はじめに

近年、携帯電話で Java アプリケーションを動作させることが一般的となってきた。これらのアプリケーションはますます高度化してきており、ピアツーピア、非同期などの通信機能が求められている。そこで、筆者らは、ピアツーピア、非同期通信機能を使用するエージェントアプリケーションを携帯電話端末や PDA(Personal Data Assistant)などの携帯端末上で動作させるための小型エージェントプラットフォームの開発を進めている。本稿では、その設計と実装について述べる。

#### 2. 携帯端末上での課題

携帯端末でエージェントプラットフォームを実現する場合、以下のような課題を解決する必要がある。

##### (1) ソフトウェアサイズの制限:

例えば、携帯電話端末では Java のソフトウェアサイズは 10KB~50KB に制限されている。従って、今後の性能向上を考慮しても、プラットフォームサイズを 100KB~200KB 程度に抑える必要がある。

##### (2) 携帯端末の特性に対する対処:

無線通信の一時的な途絶に対しても安定した通信が行えることが必要である。また、電源断や電波が届かない等による、比較的長期間の通信途絶の間も、ある程度独立してエージェントを管理する機能が必要となる。

##### (3) スケーラビリティ:

特に携帯電話端末に対しては、多数の端末上に存在する多数のエージェントに対して、通信サービスや管理サービスを効率的に提供することが必要となる。

#### 3. FIPA のエージェントプラットフォーム

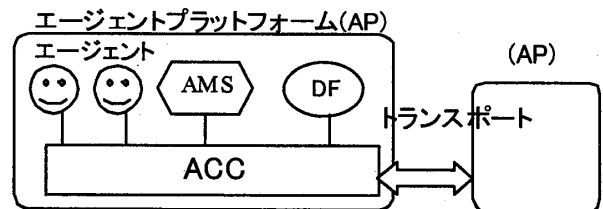
FIPA(Foundation for Intelligent Physical Agents)[1]はエージェント通信技術の標準化を進めるための非営利団体である。FIPA のエージェントプラットフォームのモデルを図 1 に示す。ACC(Agent Communication Channel)はトランスポート(論理的な通信路)を用いてエージェント間のメッセージを交換するメッセージルータである。AMS(Agent Management System)と DF(Directory Facilitator)はそれぞれエージェントの管理と、それが提供するサービスに関するエラーページサービスを提供する。

これまでに、JADE[2]、FIPA-OS[3]など FIPA 準拠のエージェントプラットフォームが開発されている。さらに、これらをベースとして、LEAP[4]、MicroFIPA-OS[5]など携帯端末を想定した小型エージェントプラットフォームが開発されてきた。LEAP は JADE の一部を PDA(PalmOS か

†(株)YRP ユビキタスネットワークワーキング研究所, YRP

‡(株)KDDI 研究所, KDDI R&D Laboratories, Inc.

WindowsCE)および PC 上の MIDP エミュレータに移植したものである。AMS、DF 等の機能は JADE に依存しており、論理的には JADE のエージェントが携帯端末上に存在する形式となる。一方 MicroFIPA-OS は PersonalJava を用いて FIPA-OS を PDA(WindowsCE)に移植したものである。AMS や DF も起動して単独でプラットフォームを構成することも可能であるが、通常は資源の制約から ACC とアプリケーションエージェントのみを起動して他の機能はサーバ側 FIPA-OS に依存する形態を取る。両者とも携帯端末側のプログラムサイズは 500KB 程度である。



ACC: Agent Communication Channel

AMS: Agent Management System

DF: Directory Facilitator

図1 FIPA エージェントプラットフォームのモデル

#### 4. 設計と実装

##### 4.1 設計の基本方針

2 節で述べた課題に対して以下のような解決方針をとった。

##### (1) ソフトウェアサイズの制限:

携帯端末上に完全なプラットフォーム機能を実装するのではなく、端末側とセンタ側に機能分散する部分プラットフォームという概念を導入し、端末上のサイズを抑制する。但し、(2)で述べるようにある程度独立した管理能力を維持するため、LEAP のようにセンタ側に完全に管理機能を依存する方式とはしない。

##### (2) 携帯端末の特性に対する対処:

無線通信の一時的な途絶に対して、メッセージのバッファをセンタ側、端末側の両方に設け、再送によりメッセージの紛失を防止する。また、全てのエージェントの状態はセンタ側で管理するが、センタと通信途絶している間も端末側で管理操作を実行可能とする。オフライン操作や異常終了によりセンタ側と端末側で状態の不一致が発生した場合に備えて、両者の状態の同期機能を設ける。

##### (3) スケーラビリティ:

アクセスが集中し性能のボトルネックとなり得るセンタ側の ACC や AMS、DF を含む部分プラットフォームを複数起動し、負荷分散を可能とする。

以上の解決方法に加えて、以下の設計方針を用いた。

##### (4) 移植性を考慮し、Java で実装する。

##### (5) 携帯電話端末で利用できる通信機能を考慮し、上り下りで別々のトランスポートを利用可能とする。

(6) 移動エージェント機能を提供する。

4.2 設計の概要

図 2 に本エージェントプラットフォームの全体構成を示す。

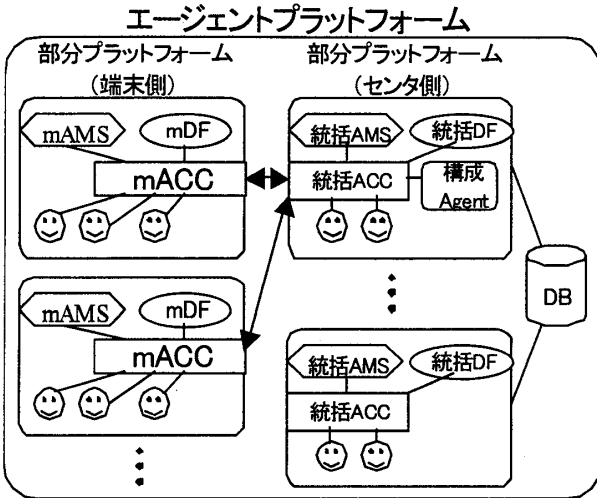


図 2 プラットフォームの全体構成

ア. センタ側・端末側の機能分担

ACC: 端末側 ACC(mACC)は、自部分プラットフォーム外のエージェントへのメッセージを特定のセンタ側ACC(統括ACC)にのみ送信する機能のみとし、小型化を行った。統括ACCは全てのルーティング機能を有する。  
 AMS: 端末側 AMS(mAMS)はエージェントの起動、停止、登録、後で述べる移動機能を提供するが、通常エージェントの管理情報はセンタ側AMS(統括AMS)が管理しており、その指示の下で動作する。統括AMSと通信できない場合のみ、単独で管理操作が行える。その情報は後で統括AMSと同期する。  
 DF: 端末側 DF(mDF)は、センタ側 DF(統括DF)に対するサービス検索のキャッシュとしてのみ動作する。

イ. スケーラビリティの実現

統括ACC、統括AMS、統括DFは管理情報やサービス登録情報をデータベースにより管理し、全てのインスタンス間で情報を共有する。構成エージェントと呼ぶ特別なエージェントを導入し、端末側部分プラットフォームの起動時に、構成エージェントがセンタ側部分プラットフォームの負荷や地域等を考慮して、mACCが接続する統括ACCを端末側に指示することで、負荷分散する。

ウ. 中断・再開機能および移動機能の実現

端末側で、部分プラットフォームを容易に停止し再開できるようにした。停止指示があると、起動中のエージェントは中断し、その状態は不揮発性メモリに格納される。再開すると、その情報から起動中のエージェントの情報が復元される。この中断・再開機能を利用してエージェントの移動機能も提供する。センタ側と端末側で使用するJava仕様が異なることから、エージェントの状態だけを転送する弱い移動機能とする。移動が指示されると、図3に示すようにエージェントの状態は記憶装置に書き出され、エージェントは中断する。移動元のAMSは移動先のAMSにその情報を転送し、移動先でエージェントに復元される。最終的に両者と(両方ともmAMSの場合)統括AMSにより情報の整合性が取られる。

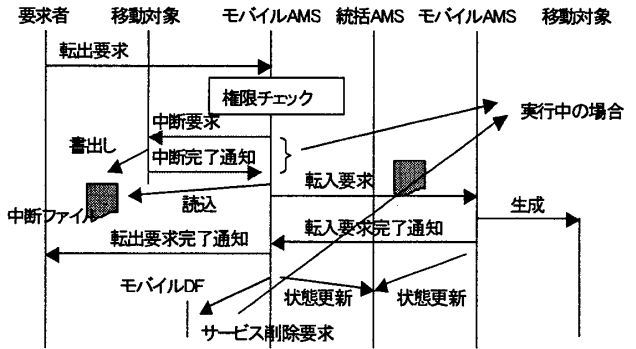


図 3 エージェント移動の流れ

4.3 実装

- ・実装では、センタ側は J2SE、端末側は J2ME+MIDP を使用した。センタ側は Linux と Windows、携帯端末側は PC 上の MIDP エミュレータでの動作を確認した。
- ・エージェント管理情報の格納は JDBC 経由とし、特定の DBMS に依存しないようにした。Postgress および Oracle8 での動作を確認した。
- ・センタ側および端末側のクラス設計を共通化し、開発規模を抑えた。
- ・端末側に専用の小型 HTTP サーバを実装し、センタ側と端末側間のトランスポートは上り下りとも HTTP を使用した。実際の携帯電話端末上では利用できるネットワーク機能やソフトウェアサイズの関係から HTTP サーバを動作できないため、下りは通信事業者に依存した方式(例えば au では C メールサービス等)とする必要がある。また、外部プラットフォームとの通信にはローザンヌ工科大が提供する HTTP 通信パッケージ[6]を利用した。
- ・J2ME にはシリアライズ機能が無いため、エージェント移動機能のために専用のシリアライズ関数を作成した。

6. おわりに

本稿では、携帯端末でエージェントアプリケーションを動作させるための、FIPA 準拠エージェントプラットフォームの設計と実装について報告した。サーバ側と端末側で機能分担しソフトウェアサイズを抑えた。通信や端末状態の変動に対処するため、バッファや端末でのエージェント管理機能を持たせた。さらに、サーバ側で複数の部分プラットフォームを動作可能としスケーラビリティを確保した。本プラットフォームの基本性能については、[7]を参照されたい。今後、他のプラットフォームとの相互接続性の確認や、応用の開発を進めていく予定である。最後に、日頃御指導頂く(株)KDDI 研究所 浅見所長に感謝します。

参考文献

[1] <http://www.fipa.org>  
 [2] Bellifemine A, 他, "Developing Multi-agent Systems with a FIPA-compliant Agent Framework", in Software- Practice and Experience, Vol.31, pp. 103-128, 2001.  
 [3] Poslad S, 他, "The Fipa-OS agent platform: Open Source for Open Standards", in Proc. of PAAM 2000, pp. 355-368, 2000.  
 [4] Bergenti F, 他, "LEAP: A FIPA Platform for Handheld and Mobile Devices", in Proc. of ATAL-2001, 2001.  
 [5] Poslad S, 他, "CRUMPET: Creation of User-friendly Mobile Services Personalized for Tourism", in Proc. of IEE 3G-2001, 2001.  
 [6] Java Agent Message Router, <http://liawww.epfl.ch/JAMR/>  
 [7] 服部他, "携帯端末のための FIPA 準拠小型プラットフォームの評価", 第 1 回 FIT 予稿集。