組み込みシステム向けエージェントフレームワーク

中溝 克明† 須田 唯之† 横山 孝典† 志田 晃一郎† 兪 明連†
†武蔵工業大学大学院 工学研究科 電気工学専攻

1 はじめに

近年，家庭内への情報機器の進出によりユビキタスネットワークが普及しつつある．それにともない，家電機器間の情報共有や協調動作により新たなサービスの実現が期待され，その手法としてエージェント技術の適用が求められている．

エージェントの動作環境をサポートするエージェントフレームワークとして，JADE[1] や FIPA-OS[2] などが提案されている．しかし，これらのエージェントフレームワークは，汎用コンピュータやワークステーションのようなリソースの豊富な環境を対象としており，家電に搭載されている容量の限られた内蔵 ROM, RAM のみで動作するような組み込みシステム上で動作させるのは困難である．

そこで本論文では，リソースに制限のある組み込みシステム上で動作可能なエージェントフレームワークを提案する．

2 分散型フレームワークアーキテクチャ

エージェントフレームワークとして必要な全ての機能を，リソースに制限のある組み込みシステム上に搭載するのは困難である．そこで，エージェントフレームワークの機能を，比較的リソースに余裕のあるホームサーバと，組み込み機器とに分散配置するアーキテクチャを採用する．

本論文で提案するエージェントフレームワークのアーキテクチャを図 1 に示す．組み込み機器側には，エージェント機能ならびにエージェント間通信を行う MTS (Message Transport System) 機能を配置する．また，ホームサーバ側に，フレームワーク内のエージェントを管理する AMS (Agent Management System)，そしてフレームワーク内において各エージェントが提供しているサービスを検索を行う DF (Directory Facilitator) を配置する．

このようなアーキテクチャを採用することで組み込み機器におけるリソース消費量を減らすだけでなく，各

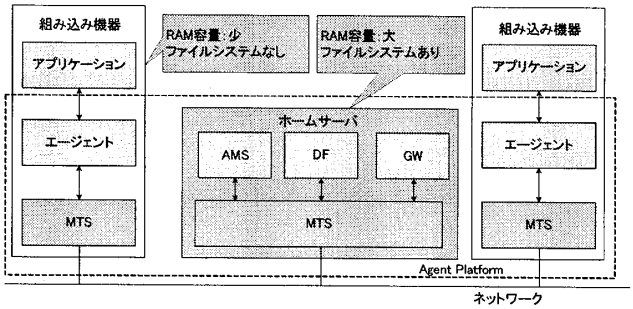

図 1: 提案エージェントフレームワークアーキテクチャ

機器上のエージェントはどのようなサービスがどの場所で提供されているかを事前に把握する必要がなくなる．各機器上のエージェントは，起動時にホームサーバ上にある AMS に登録処理を行う．AMS は登録承認を行い，DF に登録要求を受け付けたエージェントの場所 (IP アドレス等) や提供しているサービス等エージェントに関する情報を記憶する．各エージェントは，登録処理を終了後は，DF を利用することで他のエージェントサービスを利用することができる．

3 エージェント間メッセージ通信

本論文で提案するメッセージ通信方式を図 2 を用いて説明する．メッセージを送信するエージェントは，送信先のエージェント名や，エージェントが提供しているサービスを指定して，メッセージを MTS を介してホームサーバに送る．ホームサーバ上の MTS は，DF を参照し，受け取ったメッセージが指定しているエージェント名やサービス名から，送信先のエージェントが存在するノードを決定し，そのノードにメッセージを転送する中継機能を果たす．

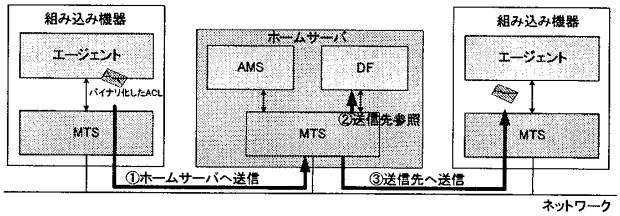

図 2: エージェント間メッセージ通信方法

エージェント間の協調動作は，ACL (Agent Communication Language) を介したメッセージのやりとりにて行われる．ACL として，FIPA (The Fundation Intelligent Physical Agents) が規定している FIPA-ACL を採用す

An Agent Framework for Emmbedded Systems
†Katsuaki NAKAMIZO, Tadayuki SUDA, Takanori YOKOYAMA, Koichiro SHIDA and Myungryun YOO
†Graduate School of Electrical Engineering, Musashi Institute of Technology

る．FIPA-ACL で規定されたエンコーディングスキームには 3 種類の方法があるが，本論文では Bit-efficient を使う．Bit-efficient を採用することで ACL メッセージの表現に必要とするバイト数が少なくなり，効率的なデータ転送ができるほか，メモリ使用量も削減できる．

本エージェントフレームワーク外のエージェントフレームワークとの通信には，Bit-efficient 以外の方法でエンコーディングしている場合も考慮する必要がある．そこで，図 1 のホームサーバにゲートウェイを配置し，エージェントフレームワーク内外の通信をサポートできるようにメッセージの変換を行う．

4 エージェントアーキテクチャ

エージェントは，その振る舞いを決定する推論部，他のエージェントとの通信を行う通信部，組み込みアプリケーションとの連携をするインターフェース，そしてそれらの動作を制御する制御部からなる．図 3 に本論文でのエージェントアーキテクチャを示す．

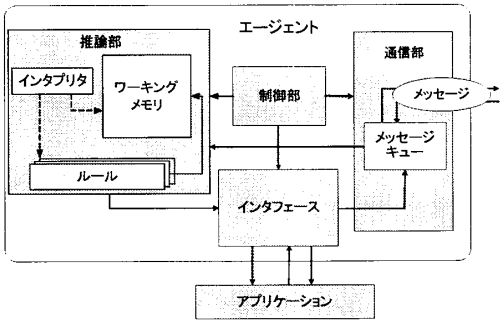

図 3: エージェントアーキテクチャ

推論部における推論アルゴリズムには，前向き推論であるプロダクションシステムを採用する．エージェントの状態はワーキングメモリの更新によって随時行われ，ワーキングメモリの状況を解釈し，ルールに記述された動作の実行や制御を行う．

エージェントの実装には C 言語を用いる．メモリサイズの制約が厳しい組み込みシステム向けの既存アプリケーションは，C 言語や C++言語で実装されていることが多い．C 言語で実装することにより，エージェントとアプリケーション間の連携効率を高めることができる．また，Java と異なり仮想マシンを必要としないので，メモリの使用量を削減できる．また，ROM 化が容易なため，組み込みシステム上に搭載するのに適している．

5 エージェントフレームワークの実装

本フレームワークを用いた，エージェントシステムの実装について述べる．エージェント設計時には，ルール・初期状態を指定した共通の記述形式に従って記述する．記述形式は，設計者にとって可読性がよいなどの観点から図 4(a) に示すような XML 形式を採用した．文法は，打矢らが開発したフレームワーク [3] の記述言語を参考に設計した．

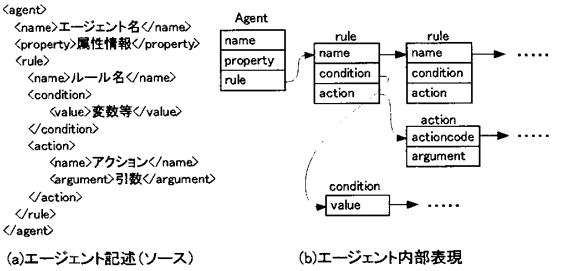

図 4: エージェント記述言語とエージェントの実装

図 5 に示すように記述されたルール等は，エージェントコンパイラにより，C 言語で書かれたエージェント本体にルールと初期状態の情報として埋め込まれる．図 4(b) にエージェントコンパイラにより生成された，C 言語によるエージェントの内部表現の構成を示す．

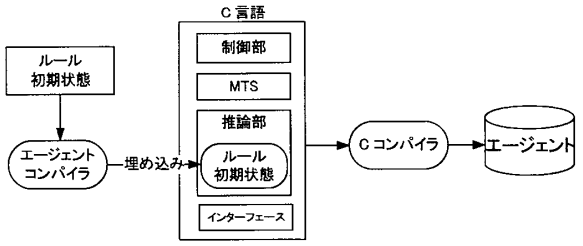

図 5: エージェントプログラム生成の流れ

なお現状では，エージェントコンパイラは完成しておらず，内部表現を直接記述し，実験・評価を行っている．

6 おわりに

本論文では，家電機器間の連携動作が可能なマルチエージェントシステムを実現することを目的に，リソースの限られた組み込みシステム上でも動作可能なエージェントフレームワークの提案をした．今後，エージェントコンパイラを完成し，様々な組み込みシステム上で動作実験を行い，本フレームワークの有効性を検証していきたい．

参考文献

[1] Giovanni Rimassa, Fabio Bellifemine, Agostino Poggi. JADE -A FIPA-compliant agent framework. *Telecom Itaria internal technical report. PAAM'99*, 1999.

[2] R.Hadingham S.Poslad, P.Buckle. The fipa-os agent platform: Open source for open standards. *PAAM2000*, 2001.

[3] 打矢隆弘, 原秀樹, 高垣暁, 菅原研次, 木下哲男. リポジトリ型エージェントフレームワークの開発と評価. 情報技術レターズ, Vol. 2, pp. 799–811, 2003.