

リソースに制限のある組み込みシステムのための エージェントフレームワーク

須田 唯之 大塚 万里子 中溝 克明 横山 孝典 志田 晃一郎 兪 明連
武蔵工業大学

1. はじめに

近年、ユビキタスネットワークの普及により、家電などによる情報共有が可能となりつつある。そこで、エージェント技術を適用し、家庭にある各家電にエージェントを搭載し、電力調整などの協調動作を行わせることが期待されている。そのためには、リソースが限られた組み込みシステムで動作するエージェントシステムを実現する必要がある。

エージェントを動作させるためには、一般的にエージェントフレームワークを利用する。エージェントフレームワークは、エージェントやエージェントシステムの設計・実装・運用を支援するための開発ツールであり、実行環境としても利用できる [1]。ところが、DASH や JADE などの既存のエージェントフレームワークは、Java 言語をベースに開発されており、豊富なメモリ容量と処理性能の高い CPU が必要となる。このため、従来のエージェントフレームワークを家電の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、家電のようなメモリ容量や CPU 性能に制限のある組み込みシステム上で動作するエージェントフレームワークを開発するのが本研究の目的である。

2. エージェントフレームワークの一般的構成

我々の提案する組み込みシステム向けエージェントフレームについて述べる前に、エージェントフレームワークの一般的な構成について説明する。図 1 はエージェントフレームワークの一般的な構成例を表している。エージェントフレームワーク上には複数のエージェントが存在し、互いにメッセージ通信を行っている。

図 1 の吹き出し内の図は、エージェントフレームワークが提供するエージェントのアーキテクチャの例である。エージェントは、推論を行う推論モジュール、通信を行う通信モジュール、外部プログラムとのやり取りを行う外部インターフェースモジュール、それらの動作を制御する制御モジュールなどから成る。

エージェントからのメッセージはエージェント本体が持つ通信モジュールからフレームワーク上の通信機構へ

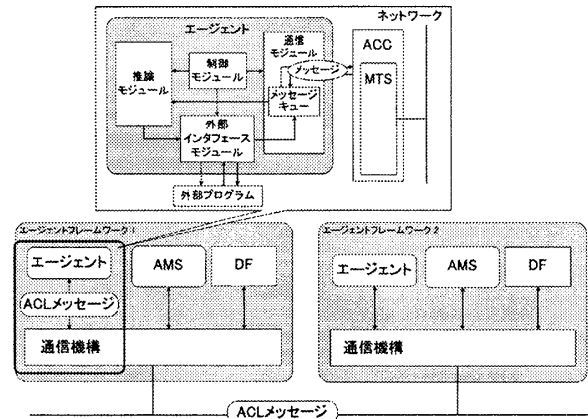


図 1 エージェントフレームワークの構成例

送られる。メッセージは ACC (Agent Communication Channel) 内の MTS (Message Transport Service) を通り、他のエージェントと通信を行う。ACC はフレームワーク内での各機能間のメッセージ通信を管理する機構である。また、MTS はエージェント間のメッセージ通信を制御する機構である。MTS を通してメッセージを送ることで、エージェント同士は存在するプラットフォームに関係なく協調した動作を行うことができる。

エージェントフレームワーク上には AMS (Agent Management Service) と DF (Directory Facilitator) が必須のものとして存在している。エージェントは、各フレームワーク上に存在する AMS によって生成から消滅までのライフサイクルを管理される。また、DF には動作しているエージェントが登録されており、各エージェントが提供するサービスやその位置などを照会する。エージェントはこれらと通信機構を介したメッセージ通信によって情報のやり取りを行う。

3. 組み込みシステム向けエージェントフレームワーク

3.1 基本方針

本研究ではリソースの限られた組み込みシステム上にエージェントフレームワークを実装するために、以下のような方針に基づいて通信機構と推論エンジンを開発する。フレームワークの機能を組み込みシステム上とホームサーバ上に分散させる。ACL メッセージは、String 形式ではなくバイナリ形式を使用する。推論方式はプロダクションシステムを採用し、推論エンジンを実装する。

また、仮想マシンを用いないため、実装言語に C 言語を用いる。これらの結果、メモリ使用量を減らし、実行速度を向上させることができる。以下、本方針に基づいた実装方式の詳細について説明する。

3.2 エージェント間通信機構

本研究では、図 2 に示すようにエージェントフレームワークの機能を組み込みシステム上とホームサーバ上に分散させる。組み込みシステム上のフレームワークではエージェントと通信機構のみを持つ。そして、エージェントを管理する AMS と DF をホームサーバ上のフレームワークに置く。

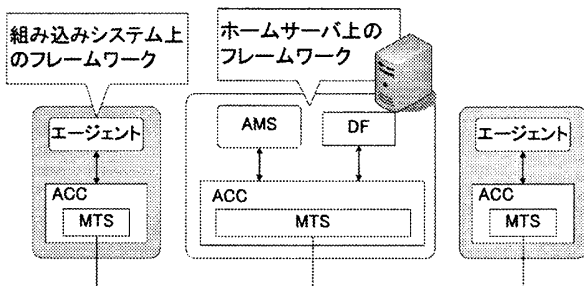


図 2 本研究でのエージェントフレームワークの構成例

AMS は、通信によってエージェントが動作可能であることを管理する。DF はエージェントの情報を保持しているため、必要となるのは他のエージェントへのメッセージを送受信する場合である。この 2 つをホームサーバ上に置くことで、それらは常に通信可能な状態となり、エージェントと同一のフレームワーク上になくとも、役割を果たすことが十分に可能となる。

エージェントがメッセージを送信する際は、送信先をホームサーバ上の DF に問い合わせる。DF は指定された機能を持つエージェントとそのアドレスを返す。エージェントは DF から送られた情報を基にメッセージを作成し送信する。

3.3 ACL メッセージ

エージェント間のメッセージは ACL (Agent Communication Language) で表現される。ACL ではエージェントに対する命令などの記述が構文として規定されている。現在一般的に使用されているのは、FIPA ACL[2] である。FIPA ACL とは、標準化団体 FIPA (Foundation for Intelligent Physical Agents)[3] によって規定されたエージェント通信言語である。

本エージェントフレームワークでは、FIPA で規定されたパラメータ 13 種のうち、メッセージ通信に最低限必要とする以下の 4 種のみを使用可能とする。

- performative メッセージの種類 (inform など 22 種)
- sender メッセージの送信者
- receiver メッセージの受信者
- content メッセージの内容

また、記述形式をバイナリ形式とする。これにより、

文字列で記述されていたパラメータを、1 パラメータ 1 バイトに収めることができる。そのため、String 形式や XML 形式で記述した場合と比べ、容量を 4 分の 1 以下に削減できる [4]。

3.4 推論モジュール

本研究では、高度な推論を要求しない家電などの組み込みシステムを対象とする。そのため、知識は if-then ルールで記述可能である。そこで、推論方式はプロダクションシステムを採用する。

プロダクションシステムを採用した場合の推論モジュールの構成を図 3 に示す。推論モジュールは、推論エンジン、ルールベース、ワーキングメモリから構成される。リソース消費量を減らすため C 言語で推論エンジンを実装する。

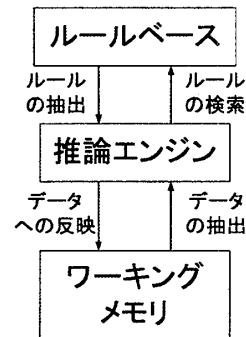


図 3 推論モジュールの構成

4. おわりに

メモリ容量や CPU 性能に制限のある組み込みシステム上で動作するエージェントフレームワークを提案した。現在その実装を進めている。実装完了後、開発した推論エンジンや通信機構の評価を行い、本エージェントフレームワーク上でエージェントアプリケーションを開発することを計画している。

参考文献

- [1] 下哲男, 桑原和宏, 菅沼拓夫, 菅原研次, 服部文夫, 原英樹, 藤田茂, エージェントシステムの作り方, 電子情報通信学会, 2001.
- [2] IPA, ACL Message Structure Specification, 2002/12/03.
- [3] IPA, <http://www.fipa.org/>
- [4] IPA AMT Envelope Representation in BitEfficient Encoding Specification, 2002/12/03.