

発展的エージェントを用いた 分散資源利用支援システムの考案*

堀内 勝 齋藤 晃 長谷川 龍弘 木村 耕
電気通信大学 情報工学科

1 はじめに

近年, 新しいビジネススタイルであるネットワークを介して, アプリケーションを提供するASP¹をはじめとし, ネットワークの資源のさまざまな利用方法が議論されている.

また, ソフトウェア開発において, エージェント, 発展機構を備えたソフトウェア [1], 適応型ソフトウェア [2] が注目を集めている.

発表者らは, かつてエージェント間通信言語ライブラリ Falcon を開発した [3][4]. Falcon ライブラリの動作は Agent-0[5] を参考にしたコミットメントルールにより決定される.

本研究では, エージェント技術を用いてネットワーク上の資源を効率よく利用する方法を検討し, Falcon ライブラリで構築されたエージェントシステムによってこれを実現する.

2 方針

システムの構築は, 以下の方針により構築する.

- ネットワーク上の多彩な機能に対応するために, 発展的機構を備えたエージェントを導入する.
- ネットワークの移動先の機能を使用するために, 適応型ソフトウェアの概念を導入する.
- これらの機能を実現するために Falcon ライブラリをベースにシステムを構築する.

3 システムの構想

本システムは図1に示すようなマルチエージェントシステムとして構成される. 以下に各エージェントの機能を示す.

* An idea of supporting system for distributed resources based on an evolutionary agent by HORIUCHI Masaru, SAITO Akira, HASEGAWA Tatsuhiko and KIMURA Koh, Department of Computer Science and Information Mathematics, The University of Electro-Communications(UEC).

¹ Application Service Provider

User Interface Agent(UIA):

ユーザとの対話を行うエージェント. Web ブラウザやメール的な役割を担う.

Evolutionary Proxy Agent(EPA):

UIA と SWA や APA の仲介役を担うモバイルエージェントであり, 基本的機能として移動機能, コミットメントルール実行機能, 移動先のメソッドの実行機能を持ち, コミットメントルールの更新による発展性を持つ.

Service Wrapping Agent(SWA):

ネットワーク上にあるサーバプログラムをラッピングし, ACL² をサーバプログラムの理解できるプロトコルに変換し, EPA とサーバプログラムとの会話を成立させる.

Application Place Agent(APA):

アプリケーションを有するネットワーク上の Server Place に常駐し, EPA にサーバ上の資源やプログラム, SWA の情報を提供する. また, EPA とアプリケーションの仲介役を担う.

Portal Place Agent(PPA):

ネットワークに散在する Server Place の情報を管理し, EPA に提供する.

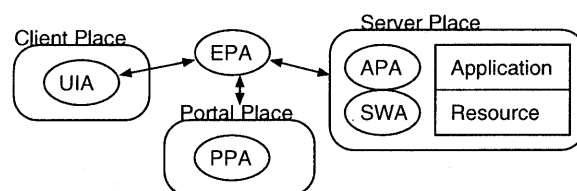


図 1: システム構成図

3.1 動作手順

1. ユーザは UIA に情報取得やアプリケーション実行の要求を出す.

² Agent Communication Language

2. EPA は, UIA の指示により, Portal Place に移動し, PPA から Server Place の情報を取得し, Server Place に移動する.
3. EPA は APA と対話することによりサーバ上の資源やアプリケーションの情報を取得し, その Place で使用できる SWA の情報を取得する.
4. EPA は SWA を介してサーバプログラムとやり取りをし, 情報を取得する.
5. EPA は情報をクライアントに持ち帰り UIA に提供する.

EPA は発展機構を備え, Server Place に移動したとき APA からコミットメントルールを取得し, 自己のコミットメントルールと結合し, その Place のメソッドを動的に取得することにより成長する. また, ユーザの要望の傾向やアプリケーションの特性等を学習する機能を有する. EPA は, Place に移動することにより, Place 特有の機能に適応することができ, Place の持つメソッドの実行が可能となる.

3.2 エージェントとサーバプログラムとの会話の実現方法

エージェントが会話を行うサーバプログラムとして, 現在 HTTP サーバプログラム, POP3 サーバプログラムを考えている. その一例として HTTP サーバプログラムの使用方法を示す.

3.2.1 HTTP サーバプログラムをラッピングする SWA の実現方法

SWA は EPA にから URL の記述された ACL を受け取る. これを HTTP サーバプログラムの解釈できる形に翻訳して HTTP サーバプログラムに転送する. SWA は HTTP サーバプログラムから情報を取得し, EPA にそのまま受け渡す. 図 2 にこの概要を示す.

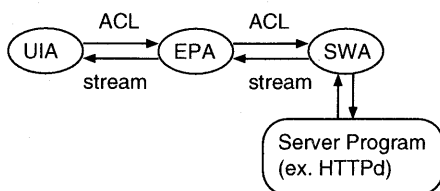


図 2: メッセージ配信手順

3.3 適応機構の導入方法

Falcon ライブラリには今回, エージェントに他の Place への移動機能及び移動先の Place のメソッドを実行する機能が追加された. 本システムではこの機能を用いて適応機能を実現する.

3.4 発展機構の導入方法

Falcon ライブラリを用いることによりコミットメントルールの更新機能を実現し, また, 動的にメソッドを追加することにより発展機構を実現する.

4 評価と考察

- 発展機構及び適応機構を実現することにより, ネットワーク上の各 Place に散在する多様な資源を利用状況に応じて獲得する一例を示すことができた.
- Falcon ライブラリを用いたことにより Falcon ライブラリの実用性が実証された.
- 標準の HTTP や POP3 を用いることにより ACL より拡張性のある会話が可能である.

5 おわりに

現在, 今回紹介した UIA, EPA, SWA のプロトタイプの実装が完成し, 構想したエージェントシステムの動作は可能となった. 今後, EPA の更なる機能拡張, 会話の充実, PPA の導入などを実現する.

参考文献

- [1] 発展機構を備えたソフトウェアの構成原理の研究, <http://www.jaist.ac.jp/~gondow/jyuten/>.
- [2] F.Kumeno, A.Ohsuga and S.Honiden: Flage: A Programming Language for Adaptive Software, 電子情報通信学会英文論文誌, Vol.E81-D, No.12, pp.1394-1403(1998).
- [3] 山腰 哲, 天田 実志, 木村 耕: “FIPA に基づくエージェントライブラリの実例”, 情報処理学会第 58 回全国大会講演論文集 (分冊 4), pp.227-228 (1999).
- [4] 酒井利治, 山腰哲, 堀内 勝, 木村 耕: “標準基盤に基づくエージェントライブラリの実装”, 情報処理学会第 60 回全国大会講演論文集 (分冊 1), pp.327-328 (2000).
- [5] Y.Shoham: “Agent-oriented programming”, Artificial Intelligence Vol.60, pp.51-92 (1993).