

ソフトウェアエージェント開発トレーニング用システム TAF の 開発と評価

原英樹[†] 木下哲男[‡] 菅原研次[†]

[†] 千葉工業大学情報ネットワーク学科
〒 275-0016 習志野市津田沼 2-17-1

[‡] 東北大学電気通信研究所 / 情報科学研究科
〒 980-8577 仙台市青葉区片平 2-1-1

{hara,suga}@net.it-chiba.ac.jp, kino@riec.tohoku.ac.jp

あらまし 我々はこれまでエージェント指向分散システムを開発するための ADIPS フレームワークの提案とこれに基づく実行環境および開発環境の開発を行ってきた。ADIPS フレームワークは柔軟なエージェントシステムを開発できるという利点を持つ反面、エージェントの概念に不慣れな技術者にとって利用することが困難であるという問題があった。本論文では、ソフトウェアエージェント教育用システム TAF の開発と評価について述べる。TAF システムは、ADIPS フレームワークの複雑な機構を簡略化して設計されており、またエージェントの動作を詳細に観察するツールを利用者に提供しているため、エージェントシステムの開発経験のない技術者でも容易に利用可能であり、ADIPS フレームワークを用いたエージェントシステムの開発手法について学習することが可能である。

キーワード エージェントシステム, 開発教育システム, ADIPS

Development and Evaluation of Training-system for Agent Framework TAF

Hideki HARA[†], Tetsuo KINOSHITA[‡] and Kenji SUGAWARA[†]

[†] Dept. of Network Science, Chiba Institute of Technology
2-17-1 Tsudanuma, Narashino 275-0016, Japan

[‡] Research Institute of Electrical Communication / Graduate School of
Information Sciences, Tohoku University
2-1-1 Katahira, Aoba, Sendai-shi, 980-8577 Japan

{hara,suga}@net.it-chiba.ac.jp, kino@riec.tohoku.ac.jp

Abstract ADIPS framework that we have proposed is a framework to provide a methodology to develop distributed systems based on an agent-oriented architecture. But this framework is hard to understand for a developer who does not have experience in development of agent systems. In this paper, we describe development and evaluation of Training-system for Agent Framework (TAF system). An architecture of TAF system is simple than ADIPS framework's architecture. Since TAF system provides a user with some useful tools to monitor and inspect behavior of agents, the user can study methods of development of agent systems even if he/she has little experience in development of agent systems.

key words agent system, training system for development, ADIPS

1 はじめに

我々はこれまでエージェント指向分散システムを開発するための ADIPS (Agent-based Distributed Information Processing System) フレームワークの提案と、これに基づく実行環境および開発環境の設計と開発を行ってきた [1][2][3][4]。しかしながら、ADIPS フレームワークは動作環境およびリポジトリと呼ばれる 2 つの場でエージェントが動作するという複雑な枠組であり、また、それぞれの場におけるエージェントの知識記述の形式および内容が異なるため知識を記述することが難しいという欠点を持つ。ADIPS フレームワークを用いて開発を行う場合、その開発者は簡単なエージェントシステムの試作を行う場合でも前述した 2 種類の知識を記述せねばならない。特にリポジトリにおけるエージェントの動作を記述するにはリポジトリ内のエージェントが行う通信のプロトコルを熟知していないと記述できないため、エージェントに不慣れた技術者が学習の過程でさまざまなエージェントシステムを試行錯誤的に開発を行う用途には不向きであった。

この問題に対して、我々はソフトウェアエージェント開発トレーニング用システム TAF (Training-system for Agent Framework) の設計と実装を行った [5]。本論文では TAF システムの設計の概要と、TAF システムを用いてエージェントシステムの開発訓練を実施した際に新たに明らかになった点について述べる。

2 TAF システムの設計

2.1 設計方針

TAF システムは、エージェントシステムや ADIPS フレームワークに関する基礎的概念の学習に利用することを目的として開発された。主な目的は ADIPS フレームワークを利用するための導入教育にある。表 1 に示したように、ADIPS フレームワークのうち初心者にとって複雑な機能や概念を省略し容易に理解できるよう設計した。以下に TAF システムの設計方針を述べる。

(1) リポジトリ機能の簡略化

ADIPS フレームワークの特徴的機能である利用者要求や動作環境の変化への適応を行う機能は、リポジトリに蓄積されたエージェントがリポジトリ内で動作することで実現されるが、これは複雑な組織構成プロトコルを用いたエージェント間通信により行われている。TAF システムでは理解しやすいように、リポジトリ内でのエージェントの動作、組織構成/組織再構成の機能を省略し、単にエージェントを蓄積する機能のみに簡略化する。

表 1: TAF システムと ADIPS フレームワークの比較

	TAF	ADIPS
リポジトリ	有	有
リポジトリ内での動作	無	有
組織構成/再構成プロトコルの処理機能	無	有
知識記述形式	ルール	仕様記述, 条件記述, ルール
計算機プロセスの制御	Java のみ	Java, C, sh, etc.
メッセージ通信の観察	可	可
エージェント内部の動作の観察	可	不可
分散システムの構築	不可	可
移動エージェント	不可	可

(2) 知識記述の簡略化

リポジトリ内でエージェントを動作させる機能を省略したため、この機能の実現に必要なエージェントの仕様記述および、組織構成/再構成プロトコルの処理に必要な条件記述を省略した。TAF システムのエージェントの知識記述はプロダクションシステムのルールの記述のみで行う。また ADIPS フレームワークのエージェントが持つ特徴的機能である計算機プロセスの制御機能は、その知識記述量を少なくするために Java プログラムの呼出機能のみに限定した。

(3) 動作観察ツールの拡張

ADIPS フレームワークで提供しているエージェント間のメッセージ通信を観察するツールに加え、ルールの発火の様子やワーキングメモリ内のファクトを観察するツールを提供し、プロダクションシステムに不慣れた利用者にも扱いやすいものにする。

(4) 分散システム機能の省略

ADIPS フレームワークでは、異なる計算機上で稼働するエージェント同士の通信や移動エージェントなどを用いたエージェントシステムの構築が可能であるが、TAF システムでは理解のしやすさのためにこれを省略した。

2.2 TAF システムの構成

図 1 に TAF システムの構成を示す。TAF システムは、リポジトリ、ワークスペースおよびエージェントモニターから構成される。個々のエージェントの知識をエージェントプログラムと呼ぶ。リポジトリとはエージェントプログラムを蓄積し、利用者から指定されたエージェントプログラムをワークスペースに送り込む機能を持つ機構である。ワークスペースとはリポジトリから呼び出したエージェントを起動して動作させたり、問題解決処理

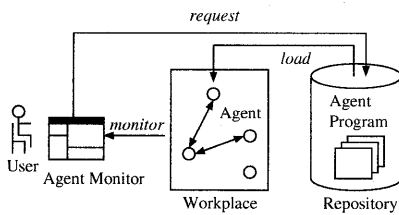


図 1: TAF システムの構成

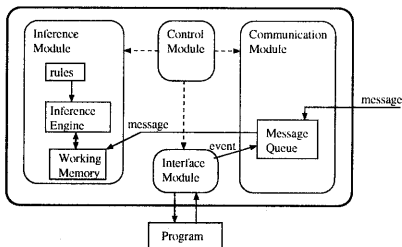


図 2: エージェントのアーキテクチャ

を実行したりする場を提供する機構である。ワークスペース上では、複数のエージェントが協調したり組織を作ることにより、エージェントシステムが構成される。エージェントシステムを構成する個々のエージェントの呼び出しや起動は、エージェント利用者がその要求をリポジトリに送ることによって行われる。エージェントモニタは、ワークスペース上に呼び出されて稼働するエージェントの動作を観察するためのツールである。利用者はエージェントモニタを用いて、エージェント間のメッセージの通信の様子、送受信したメッセージの履歴の蓄積と表示、エージェント内部のルールを選択とルールのアクションの実行、ワーキングメモリの内容など、エージェントの動作や状態を観察する。

ワークスペース上で動作するエージェントのアーキテクチャを図2に示す。通信モジュールはエージェント相互間のメッセージ通信を担当するモジュールであり、推論モジュールはエージェントプログラムに従い問題解決処理を実行するためのプロダクションシステムである。また外部インタフェースモジュールはエージェントから外部のプログラムを起動しそれを呼び出すためのモジュールである。これらのモジュールは制御モジュールにより、他のエージェントからのメッセージあるいは外部のプログラムからの状態通知を受信すると、推論モジュールが動作するよう制御されている。

エージェント間で送受信されるメッセージの形式を

```
( <performative>
  :from <sender>
  :to <receiver>
  :content <oav> )
```

図 3: エージェント通信言語

```
(agent Simple

(property
  (create :author "太郎@ieice" :date "Apr.15,2000"))

(initial_facts)

(knowledge

  (rule accept
    (message :performative request-information
      :from ?from :content (
        question :sentence "Designer?")) = ?msg

    (create :author ?author)
    -->
    (send :performative inform
      :to ?from
      :content (answer :sentence ?author))
    (remove ?msg))

  (rule deny
    (message :from ?from :content ?content) = ?msg
    -->
    (send :performative sorry
      :to ?from
      :content ?content)
    (remove ?msg))

) )
```

図 4: エージェントプログラムの例

図3に示す。ただし、<performative>はメッセージの種別を表す文字列、<sender>はメッセージを送信したエージェント名を表す文字列、<receiver>はメッセージを送信したエージェント名を表す文字列、<oav>は内容を表す文字列および、属性と値の組で表すメッセージの内容である。

2.3 エージェントプログラム

エージェントプログラムはルール形式で記述する。利用者はこの知識記述形式についての知識が必要である。ただし、プロダクションシステムに不慣れた利用者を考慮して記述形式や処理方法はできるだけ簡略化している。例えば競合解決戦略などは省略し、ルールは記述順に first-match で発火するものとする。

エージェントプログラムの例を図4に示す。エージェントプログラムは、(1) 予約語 agent, (2) エージェント名 name, (3) エージェント固有の情報を記述した属性記述リスト (property ...), (4) エージェントの初期

状態の記述リスト (facts ...) および, (5) エージェント動作知識の本体を記述する知識記述リスト (knowledge ...), の5つの要素からなる. 図4に示したエージェントプログラムは, エージェントの設計者を問い合わせるメッセージ

```
(message :performative request-information
      :content (question :sentence "Designer?"))
```

を受信したときに, 属性記述として記述したエージェントの設計者の名前を返す機能を持つ.

3 TAF システムの利用

3.1 学習手順

TAF システムにはエージェントシステムの基礎的概念を学習するための例題が組み込まれている. 各例題は, (1) エージェントシステムを説明する文書, (2) エージェントシステムを構成するエージェントプログラム, (3) 機能の変更や拡張を行う演習問題, の3つから構成される. 利用者は個々の例題について, まず文書を読むことでその概要を把握し, 次にシステムに組み込まれた例題を動作させ, 最後に演習問題を行う, という手順でエージェントシステムの学習を行う. 例題の構成を表2に示す. 表中で, 例題内容は利用者に提供する学習項目を表し, 個数はその例題で出現するエージェントの個数を表す.

例題 (1)(2)(3) では他のエージェントと通信を行わないエージェントを用いて, TAF システムの利用方法やエージェントの基本的概念についての学習を行う. 例題 (1) の目的は, TAF システムの操作方法とエージェント通信言語の概念を学習することにある. 既出の図4はこの例題で利用するエージェントのプログラムである. 例題 (2) では, エージェントに簡単なデータを持たせ, エージェント通信言語による利用者からの問い合わせに回答できるエージェントプログラムを作成する. 例題 (3) では, ADIPS フレームワークが持つ重要な概念の一つである既存プログラムのエージェントによる制御について学習する.

例題 (4)(5)(6) ではエージェント間の通信を行う演習を行う. 例題 (4) で初めて他のエージェントと通信を行うエージェントを学習する. 例題 (5) では, 利用者からの要求を受け取り小問題に問題分割して他の3つのエージェントに分配するという, 1対多の通信を行うエージェントシステムの開発を行う. 例題 (6) では, 会議の予約をスケジューリングするエージェントを2つ作り, 予約状況や優先順位に従いもう一つのエージェン

表 2: 例題の構成

例題内容	個数
(1) エージェントとの対話	1
(2) エージェントへの問い合わせ	1
(3) 外部プログラムの制御	1
(4) エージェントによる対話の支援	2
(5) エージェントによる問題分割	4
(6) 状況の認識と判断	3
(7) 契約ネットプロトコルを用いた協調	4
(8) 黑板モデルを用いた協調	8
(9) 実用的なシステムへの応用	7

トからの予約を受理できるかどうかの判断を行わせるエージェントシステムについて学ぶ.

例題 (7)(8)(9) ではこれまでの例題よりも複雑な協調を行うエージェントシステムの学習を行う. 例題 (7) では ADIPS フレームワークの組織構成/再構成プロトコルの導入教育として, これらのプロトコルが基本としている契約ネットプロトコル [6] による協調について学習する. 例題 (8) では黑板モデルに基づきメッセージ交換を行うエージェントシステムの学習を行う. 例題 (9) ではこれまでの例題を組合せ実用的なシステムを構築する.

3.2 利用方法

TAF システムを起動すると, エージェントモニタと呼ぶインタフェースが表示される (図5). 利用者はこのエージェントモニタを介して後出するエージェントインスペクタやメッセージログビューアなどの起動を行う. エージェントモニタは左上部, 左下部, 右上部, 右下部の4つのサブウィンドウからなる. 左上部には, ワークスペース上で動作しているエージェントの一覧をアイコン形式で表示する. 各アイコンの隣には, エージェントの状態が文字列で表示される. エージェントがメッセージを送受信する場合には, 送信元のエージェントのアイコンから送信先のエージェントのアイコンへ矢印が描画される. このときメッセージの内容は, 左下部に表示される. 利用者がエージェントにメッセージを送信する場合には, 右下部の ACL エディタに, メッセージのパフォーマティブ, メッセージを送るエージェント名およびメッセージの内容を記述する. 右上部には, TAF システムから利用者に対し提供される情報や, エージェントから利用者へ送信されたメッセージが表示される.

エージェントモニタ上に表示されたエージェントのアイコンをマウスでクリックすることで, エージェントインスペクタと呼ばれるツールが起動する (図6). エージェントインスペクタは, ルールリスト (右上部), プ

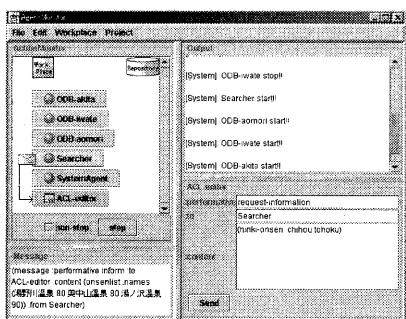


図 5: エージェントモニタ

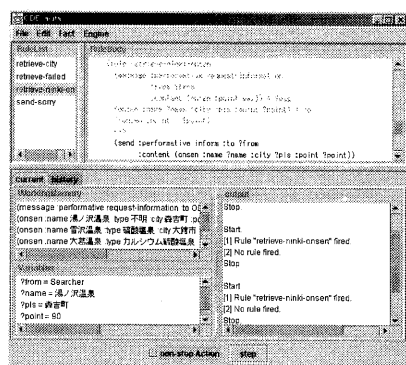


図 6: エージェントインスペクタ

ログラムビューア（左上部）および動作モニタ（下部）の3つのサブウィンドウからなる。ルールリストにはエージェントプログラムに記述されたルールのルール名一覧が表示される。プログラムビューアにはエージェントプログラムが表示される。これはルールリストと連動しており、ルールリストでルールを選択するとそのルールが表示される。また、エージェントがルールのアクションを実行しているときには、実行しているアクションを反転表示させる機能を持つ。動作モニタはcurrentタブとhistoryタブのそれぞれの選択状態により表示内容が変わる。currentタブを選択している場合、ワーキングメモリ内のファクトの一覧、ルールに記述された変数の内容、発火したルール名が表示される。またhistoryタブを選択した場合、発火したルールの履歴と発火後のワーキングメモリの内容が表示される。

個々のエージェントのメッセージの送信内容の確認には、図7に示したメッセージログビューアと呼ばれるツールを用いる。このツールはワーキングメモリで動作しているエージェントが送受信したメッセージの履歴を表示する。対象とするエージェントは上部にあるエージェントリストから選択する。Sender/Receiverボタンで選択したエージェントが送信/受信したメッセージの内容を下部に表示する。

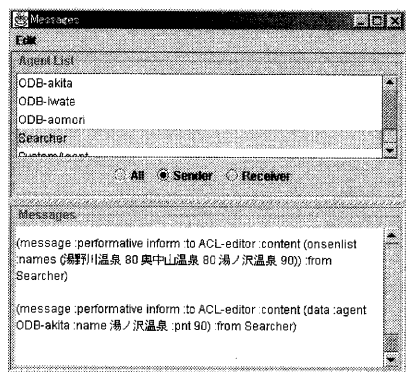


図 7: メッセージログビューア

4 評価

TAF システムの評価実験として、本学の学部4年生の学生7人にTAFシステムを実際に利用してもらい、エージェントシステムの開発訓練を行った。評価実験は次のような学生を対象として行った。

- ADIPS フレームワークを含むエージェントシステム開発経験はない。また、エージェントに関する概念を持たない。

- プロダクションシステムを利用した経験はない。また、ルールやファクト、ワーキングメモリなど、プロダクションシステムに関する概念もほとんど理解していない。

- 学部の講義で演習を行った他は、プログラミング言語によるソフトウェア開発経験はない。

評価実験は3.1で述べた学習手順に基づいて、講義および演習形式で行った。1回の講義および演習の時間は1.5時間程であり、4週間に渡って4回行った。各回の進め方は、表2で示した例題のうち1つまたは2つについて、まず例題の説明文書を用いて講義を行い学習内容を理解させ、次に演習として、例題として用意したエージェントシステムを起動しその動作の様子の観察を行い、続いて演習問題を解答させるという方法で行った。例題の終了後、エージェントシステムやプロダクションシステムに関する理解度を調査するためにアンケートを実施した。アンケートの集計結果から得られたTAF

表 3: エージェントシステムに関する理解度

例題 番号	エージェントの 個数	作成可能と 回答した人数
(1)	1	6
(2)	1	7
(3)	1	5
(4)	2	0
(5)	4	0
(6)	3	0
(7)	4	-
(8)	8	-
(9)	7	-

システムの評価を以下に述べる。

(1) エージェントシステムの理解に対する効果

エージェントシステムに関する理解度を調べるために、「例題とおなじようなエージェントシステムは開発できるか?」という質問事項をアンケート内に含めた。この質問に対し作成可能と回答した人数を表3に示す。例題(7)(8)(9)については、今回は評価実験を行わなかった。エージェントの個数が1つである例題(1)(2)(3)については半数以上が作成可能と回答し、エージェントの個数が複数である例題(4)(5)(6)が不可能であると回答した原因は、対象となった学生がプロダクションシステムに不慣れであることが原因と考えられる。また、例題(4)(5)(6)はエージェント間でメッセージ通信を行うため、例題(1)(2)(3)と比較してエージェントプログラムのサイズが大きく、難易度が急激に増加することも理解しにくい原因と考えられる。

(2) プロダクションシステムの理解に対する効果

アンケート中で、ルール、ファクトなどプロダクションシステムで用いられる用語についての理解度をTAFシステムの使用前と使用後に分けて調査した。その結果、使用前に比べて使用後の理解度が向上していることが分かった。これは評価実験中にプロダクションシステムに関する講義を行ったこともあるが、アンケートの回答にはルール発火やワーキングメモリを観察するツールの有効性について言及したものも見られた。このことから、TAFシステムは基本的なプロダクションシステムの概念を理解していればプロダクションシステムの演習を行うことが可能であることが判明した。また、エージェントの動作を観察するツールの有用性が明らかになった。

5 まとめ

ソフトウェアエージェント開発教育用システムTAFの設計および、設計に基づいて実装したシステムにつ

いて述べた。TAFシステムの目的は、エージェントの概念に不慣れな技術者がADIPSフレームワーク上で動作するエージェントシステムの開発のトレーニングを行うツールとして利用することにある。システムの特長として、ADIPSフレームワークの構成や知識記述言語を簡略化し、またエージェントの動作を把握するためのツールを提供することで、エージェントシステムの開発経験がない技術者が容易に利用できる点が挙げられる。評価実験の結果、プロダクションシステムに関する知識や利用経験がない場合には利用が難しいが、プロダクションシステムの学習にも利用できることが判明した。現在、TAFシステムでプロダクションシステムの学習を行った学生を対象として、TAFシステムを用いたエージェントシステムの開発のトレーニングの評価を行っている。

参考文献

- [1] 藤田 茂, 菅原研次, 木下哲男, 白鳥則郎: 分散処理システムのエージェント指向アーキテクチャ, 情報処理学会論文誌, Vol. 37, No. 5, pp. 840-852 (1996).
- [2] 原 英樹, 藤田 茂, 菅原研次, 木下哲男, 白鳥則郎: 計算機プロセスのエージェント化のための知識記述方式, 電子情報通信学会論文誌, Vol. J81-D-I, No. 5, pp. 566-573 (1998).
- [3] 藤田 茂, 原 英樹, 菅原研次, 木下哲男, 白鳥則郎: エージェント指向分散処理システムADIPSのための組織構成エージェントの領域知識記述形式, 情報処理学会論文誌, Vol. 39, No. 2, pp. 188-198 (1998).
- [4] 原 英樹, 藤田 茂, 菅原研次, 木下哲男, 白鳥則郎: ADIPSフレームワークのためのエージェント開発支援環境, 情報処理学会論文誌, Vol. 40, No. 2, pp. 4030-4040 (1999).
- [5] 原 英樹, 今野 将, 菅原研次, 木下哲男: ソフトウェアエージェント開発教育システムTAFの設計と実装, ソフトウェアとその応用特集ワークショップ(SAA2000)講演論文集, pp. 183-190 (2000).
- [6] R.G.Smith: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, IEEE Transactions on Computers, Vol. 29, No. 12, pp. 1104-1113 (1980).