

プロダクションモデルを利用した ADIPSエージェント記述言語の拡張

分根享[†] 原英樹^{††} 藤田茂[†] 菅原研次^{††} 木下哲男[‡] 白鳥則郎[‡]

[†]千葉工業大学情報工学科

〒275-0016 習志野市津田沼2-17-1

^{††}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

〒980-8577 仙台市青葉区片平2-1-1

あらまし

我々はこれまでエージェント指向分散処理システムを実現するための枠組みとしてADIPSフレームワークの研究開発を行ってきた。ADIPSフレームワークでは宣言的な記述言語として定義されたエージェント記述言語ADIPS/Lを用いて分散システム設計知識の記述を行っている。しかし、この枠組みを用いてネットワーク上での包括的な作業支援を提供する人間-エージェント共生空間を実現するためには、多数のエージェントや人間との協調と知識処理を実現する必要があり、宣言的知識記述だけでは実現が困難であった。本稿ではエージェントの知識処理機構に対してプロダクションモデルを導入し、エージェント記述言語ADIPS/Lの拡張を行うことで人間-エージェント共生空間で利用されるエージェントの作成を容易にした。

キーワード エージェント エージェント指向システム 人間-エージェント共生空間

The extension of the ADIPS agent description language based on the production model

Toru Bunne[†] Hideki Hara^{††} Shigeru Fujita[†] Kenji Sugawara^{††}

Tetsuo Kinoshita[‡] Norio Shiratori[‡]

[†]Dept. of Computer Science, Chiba Institute of Technology
2-17-1, Tudanuma, Narashino 275-0016

^{††}Dept. of Network Science, Chiba Institute of Technology
2-17-1, Tudanuma, Narashino 275-0016

[‡]Research Institute of Electrical Communication, Tohoku University
2-1-1, katahira, Aoba-ku, Sendai 980-8577

Abstract

We proposed the Agent-based Distributed Information Processing System; ADIPS. An ADIPS Agent is written by ADIPS/L that is declarative description. Many agents and humans exist in the Human-Agent Symbiotic Space, and cooperate each other. But it is difficult to write the knowledge for cooperative knowledge description. In this paper, the extension of our agent model is shown to be designed and write program of agents in the Human-Agent Symbiotic Space.

key words Agent, Agent-based Computing System, Human-Agent Symbiotic Space

1 はじめに

オブジェクト指向プログラミングモデルを発展させた新しいプログラミングモデルとして、ソフトウェアエージェント [1-3] の研究が行われている。我々はこれまでにソフトウェアエージェントを構成単位して分散システムを自律的に調整/構成/再構成することで利用者要求を充足し続ける Agent-based Distributed Information Processing System: ADIPS の研究開発を行ってきた [4]。ADIPS を実現するソフトウェアエージェントは、分散システムを構成するモジュールの設計知識と、ADIPS で定義された分散システム組み上げのためのプロトコルに基づいて、利用者要求を充足する分散システムを構成し、計算機環境変化、利用者要求変化に対して必要な動作を実現する。

ADIPS を実現する各々のソフトウェアエージェントへの設計知識の記述は宣言的な記述言語として定義された ADIPS/L により行われる。ADIPS を実現するソフトウェアエージェントの動作は、計算機環境の変化、利用者要求を検出し、これに対応するための動作を導く設計知識を処理するために、リアクティブエージェント的なものであった。ADIPS は分散システムを自律的に構成し利用するという目的が中心であったために、構成された ADIPS 間の協調、特に異なる目的の分散処理システムを実現しているもの同士の協調は ad hoc な設計者の知識記述に依存し、設計者の意図しない協調を実現することは困難であり、またその記述を ADIPS/L で行うことは設計者の負担となっていた。人間の作業を支援するソフトウェアエージェントを ADIPS フレームワーク上で行おうとする場合に、上述の問題点がソフトウェアエージェント知識記述の問題となっていた。

本稿では異なる利用者要求を充足するために実現された ADIPS 間の協調を実現するために、これまで ad hoc に実現されてきた協調プロトコルの形式的な定義を与えることを目的として、ADIPS を実現するソフトウェアエージェントの拡張を行い、ADIPS/L の拡張定義を行う。この拡張はプロダクションモデルをソフトウェアエージェントの知識記述に導入することで実現した。以下、2 章で ADIPS の概念について述べ、3 章で ADIPS により実現される人間-エージェント共生空間の概念

について説明し、4 章でプロダクションモデルによる拡張について述べ、5 章でプロダクションモデルを用いた例を示し、6 章でまとめを述べる。

2 ADIPS フレームワーク

ADIPS フレームワークは分散システムを、ソフトウェアエージェントにより構築するための枠組である [4]。ADIPS フレームワークの概念図を図 1 に示す。ADIPS フレームワークは、利用者要求を受け取り目的の分散システムを構成する複数のソフトウェアエージェントの動作の場であるリポジトリと、構成された分散システムを制御監視するソフトウェアエージェントの動作の場である動作環境から構成される。現在、ADIPS フレームワークの実装として、C++ によるものと Java によるものが存在するが、本稿では Java による実装を例に述べる。

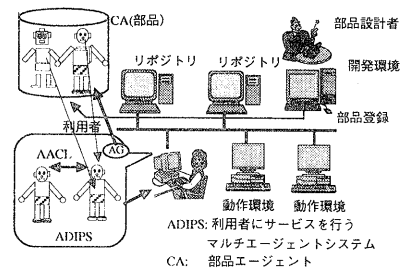


図 1: ADIPS フレームワーク

ADIPS フレームワークに基づき分散システムを実現するソフトウェアエージェントは、計算機環境の変化 (例えば、ビデオ会議システム利用時のネットワーク帯域変化)、利用者要求の変化 (たとえば、画面サイズ変更) に対して、分散システムの動作パラメータを変更する、分散システムに機能を追加するために新たなソフトウェアエージェントを追加する、分散システムの機能を大幅に変更するために分散システムの再構成を行うといった動作を、あらかじめ記述された設計知識に基づいて実現する。これらソフトウェアエージェントの動作により、ADIPS フレームワークは以下の 4 つの特徴を持つ。

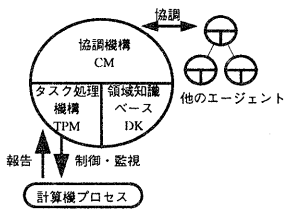


図 2: ADIPS エージェントモデル

- 利用者要求駆動による自動システム構成。
- 障害発生などのイベント駆動による自律的システム再構成。
- 自律的システム構成・再構成を目的としたエージェントによる設計者知識利用。
- エージェント化により既存分散システムを構成する計算機プロセス/ソフトウェアモジュールの系統的な再利用。

計算機プロセス/ソフトウェアモジュールのエージェント化とは既存分散処理システムの構成要素である計算機プロセス/ソフトウェアモジュールに対して、エージェントによる制御監視機構を追加し、設計知識に基づく自律的動作を可能にすることを示す。

上述の特徴はそれぞれのソフトウェアエージェントが、分散システム構成要素の設計知識をもって動作することで実現される。分散システム設計知識はエージェント内部の領域知識ベース (Domain

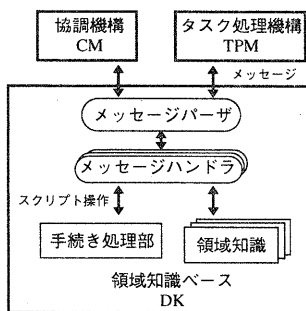


図 3: 領域知識ベース

Knowledge-base:DK) により利用される。リポトリ上で、利用者要求を充足する分散システム構築のために用いられる ADIPS 構成プロトコル (ADIPS Assembly Protocol¹) は、協調機構 (Cooperation Mechanism: CM) により処理され、DK 上の知識を利用して利用者要求を充足する分散システムを構成する。分散システムを実現する個別の計算機プロセスの制御監視は、タスク処理機構 (Task Processing Mechanism: TPM) により行われる。このエージェントのモデルを図 2 に、領域知識ベースの内部モデルを図 3 に示す。ある利用者要求を充足する分散システムは、これらの機能要素からなる複数のソフトウェアエージェントと計算機プロセスにより実現されるエージェント組織として、リポトリ上で設計され動作環境上に動的に生成される。

3 人間-エージェント 共生空間

本章では ADIPS フレームワークの利用例である、人間-エージェント 共生空間の概念について述べる。人間-エージェント 共生空間とは現実社会の人間とネットワーク環境上のエージェントが互いに協調して、ネットワーク社会における人間の社会活動を支援する仮想空間概念である [5]。共生空間には複数の異なる目的をもった利用者が同時に参加し、複数の分散システムが動作することから、人間-人間、人間-エージェント、エージェント-エージェント間で多様な協調作業が行われる。人間-エージェント 共生空間は図 4 に示すシステム構成によって実現される。

人間-エージェント 共生空間上で利用されるエージェントとしては、利用者要求を受け付けるインタフェースエージェント、共生空間におけるタスクの管理を行うタスク管理エージェント、タスクの処理を行うタスク処理エージェント等がある。ここでタスクとは“ある目的を実現するための計算機プロセスによる処理”を示す。具体的には数値処理、データベース、知識ベース、視覚化などの処理である。

人間-エージェント 共生空間システムは共生空間インタフェースシステムと ADIPS によって実現す

¹従来 ADIPS フレームワーク [4] では、“拡張契約ネットワークプロトコル”として定義していた。

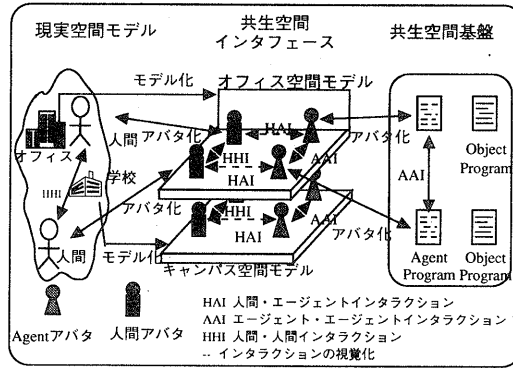


図 4: 人間-エージェント共生空間モデル

る共生空間基盤システムによって構築される。現実空間モデルとは、人間-エージェント共生空間が支援の対象とする現実世界における人間の活動や、必要な情報資源や処理を形式化したモデルである。共生空間インタフェースは、人間の間の協調や人間とエージェント間の協調を実現するさいに必要な仮想空間やアバタ等のビューモデルとインタラクションを行う際に用いられるダイアログモデルの集合である。共生空間基盤とは、人間-エージェント共生空間において人間にサービスを行い、人間の活動を協調的に支援するソフトウェアシステムの集合である。エージェントプログラムの集合とオブジェクトプログラムと呼ばれるデータや知識の集合及び、共生空間インタフェースにおける相互作用規約により構成される。共生空間に存在するエージェントには、インタフェースエージェント、タスク管理エージェント、通信サービス利用エージェント、ネットワーク監視エージェント、情報管理エージェント、情報検索エージェントなどが挙げられる。これらのエージェントは人間やエージェントと協調して共生空間で目的のタスクを実行する。

4 プロダクションモデルによる拡張

共生空間上では、人間とソフトウェアエージェント、ソフトウェアエージェント同士の多様な協調作業処理を実現する必要がある。設計知識記述を主な目的にして来た ADIPS/L で、これら協調作業処理を記述するのは困難な作業であり設計者の負担となっていた。そこで、領域知識ベースに対して設計知識だけではなく、協調作業で必要とされる協調プロトコル処理を可能な拡張を行った。

図 3 で示した領域知識ベースを拡張した領域知識ベースの概念図を図 5 に示す。

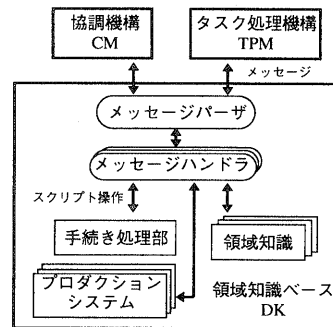


図 5: 拡張領域知識ベース

拡張された知識処理の手順を以下に示す。

1. エージェント知識記述の中に、プロダクションシステムによる処理を必要とする箇所が ADIPS/L のなかに出現する。
2. ADIPS/L から出現箇所が切り出され、プロダクションシステムによる処理が依頼される。
3. プロダクションシステムは、あらかじめ記述されたルールセットを用いて推論を行う。
4. プロダクションシステムの推論結果が ADIPS/L の表現に変換される。
5. 変換された ADIPS/L による処理が行われる。

更に ADIPS/L の記述としてプロダクションシステムが利用するルール集合を動的に変更することで、計算機環境、利用者要求、他のエージェントに対して適切な知識を表現したルール利用を行っている。

これまで様々な言語上で知識処理系を呼び出すための研究 [6] が行われているが、本研究では、プロダクションモデルを導入するために CLIPS (C Language Integrated Production System) [7] を用いた。CLIPS はルールベースの知識記述言語処理系であり、複数の言語/プラットフォーム上での実装が存在する。本研究では、ADIPS の実装が Java で行われていたことから、CLIPS の Java による実装である Jess (the Java Expert System Shell) [8] を用いた。

5 プロダクションモデルを用いた事例

本章では人間-エージェント共生空間上のエージェントにおいて、プロダクションモデルがどのように用いられるのかを述べる。

利用者からインタフェースエージェントを通じて図 6 に示される共生空間のエージェントであるタスク管理エージェントに対して次のような要求が伝えられたとする。

「デジタルカメラを買いたいのだが、どれを買ったら良いのか分からないため、その比較資料を作って欲しい。」

この依頼内容を実現するために、タスク管理エージェントは、利用者要求をいくつかのタスクに分

割するための推論を行う。この推論の際に用いられる ADIPS/L を図 7 に、また呼び出される知識セットの一部を図 8 に示す。

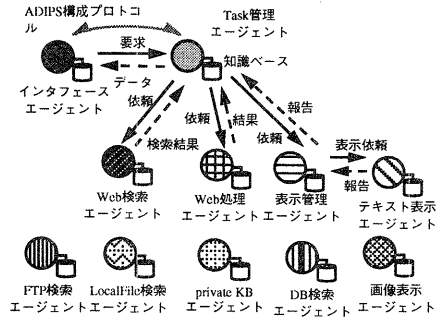


図 6: 人間-エージェント共生空間のためのエージェント例

定型の検索要求タスクは、検索エージェントへの検索依頼タスクと、表示管理エージェントへの表示依頼の 2 つのタスクに分割される。しかし今回は利用者要求を充足するためには、比較のためデータを提示する必要があるため、定型の分割では不十分である。そこで、複数の製品の情報を獲得し、それら製品を同一の基準で比較する作業を付加したタスク分割方式を推論により導き、以下のタスクを実行する系列を生成する。

- Web 検索エージェントにデジタルカメラについての検索依頼を行う。
- Web 処理エージェントに検索結果の比較を依頼する。
- 表示管理エージェントに結果を表示させる。

上記推論結果に従い、まずタスク管理エージェントはウェブ検索エージェントにデジタルカメラについての検索依頼を行う。依頼を受け付けたウェブ検索エージェントは、どの URL から検索を行ったらよいかについて推論を行い、商品についてのデータであることから、そのデータが得られそうなウェブページから検索を行うことを決定する。ウェブ検索エージェントは、推論結果に基づき、ホームページから情報の収集を行い、検索結果をタスク管理エージェントに対して返答する。

タスク管理エージェントは検索結果をウェブデータ処理エージェントに比較してもらうように、ウェブ処理エージェントは複数のページからキーワードの抽出を行い、一つのテキストファイルに変換する。変換したテキストファイルをタスク管理エージェントに渡す。タスク管理エージェントは受け取ったテキストファイルを表示管理エージェントに示し、利用者端末上での表示依頼を行う。表示管理エージェントはこのテキストデータの内容を抜えるエージェントもしくはアプリケーションが無いかわ推論を行い、テキストファイルを表示可能な計算機プロセスを制御可能なエージェントに表示依頼を行い、利用者端末上での表示が実現される。

```
(interaction inte-1 (
  (msg ask
    :from ?from
    :content (
      (task ?task)
      (content ?content)))
  (do (
    (eval clips task-division
      :args(task ?task content ?content)
      :bind(fact ?fact))
    (eval pnut send-ask
      :args(fact ?fact)
      :bind(result ?result))))))
```

図 7: ADIPS/L による記述例

```
(defrule task-search1
  (task search)
  =>
  (assert (AgAction requestSearch))
  (assert (AgAction showResult)))
(defrule task-requestSearch1
  (AgAction requestSearch)
  (field merchandise)
  =>
  (serachMedia web))
```

図 8: CLIPS による知識記述例

6 まとめ

本稿ではこれまで分散システム構築を主な目的として来た ADIPS に対して、人間-エージェント共生空間で必要とされる協調作業を実現するためのプロダクションモデルを用いたエージェント内

部の領域知識処理機構の拡張とエージェントの知識記述言語 ADIPS/L の拡張について述べた。今後、人間-エージェント共生空間での協調プロトコルの定義とこれを処理する協調知識の設計を行う予定である。

参考文献

- [1] 服部文夫: エージェント言語, コンピュータソフトウェア, Vol.14, No.4, pp.335-344(1997)
- [2] 富士通: Java 用マルチ・エージェント記述ライブラリ, <http://www.fujitsu.co.jp/hyper-text/free/kafka/jp/>, (6/25/1998 現在)
- [3] 東芝: Plangent, <http://www2.toshiba.co.jp/plangent/index.j.htm>, (6/25/1998 現在)
- [4] 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)
- [5] 今野将, 原英樹, 藤田茂, 菅原研次, 木下哲男, 白鳥則男: 拡張現実空間概念に基づくオフィスワーク支援環境の試作, 情報処理学会研究会報告 (マルチメディアと分散処理研究会), Vol.98, No.15, pp.19-24(1997)
- [6] Practical Development of Internet Prolog Applications using a Java Front End, <http://clement.info.umoncton.ca/pub/lp-net/proceedings97/beltagy/beltagy.html>, (6/25/1998 現在)
- [7] CLIPS A Tool for Building Expert System, <http://www.ghg.net/clips/CLIPS.html>, (6/25/1998 現在)
- [8] JESS, the Java Expert System Shell, <http://herzberg.ca.sandia.gov/jess/>, (6/25/1998 現在)