

協調問題解決型マルチエージェントシステム構築基盤とその評価

澤本 潤[†]、辻 秀一^{††}、小泉寿男^{†††}

エージェントは、社会的な営みをネットワーク上で実現することが可能な自律性、社会性等をもったソフトウェアシステムとして研究開発成果が期待されている。しかしながら、エージェント技術の現状での研究は、特定の応用領域を対象にした研究が多く行われており、得られた結果が一般性に乏しく、既存の研究の枠組みが一般的には他の分野には適用しにくいといったレベルに留まっている。また、実用面でもシステムの開発方式の整備が不十分、実用問題への適用事例が少ないといった未成熟な状況であると考えられる。本研究では、こういった背景を踏まえ、まずエージェントによるアプリケーション構築基盤として、マルチエージェントによる協調問題解決方式の提示を行う。さらに、開発支援手法としてエージェントモデルとタスクモデルによるテンプレートを構築し利用する方式について述べ、計画問題やネットワークコミュニティ支援の分野での応用構築実験による有効性の検証について考察を行う。

Multi-Agent System Development Framework for Cooperative Problem Solving and its Evaluation

Jun Sawamoto[†], Hidekazu Tsuji^{††}, Hisao Koizumi^{†††}

The research and development results of agent technology are awaited as the software solution with autonomy and social ability, which can provide the way to realize social activities on the Net. However, the current status of the agent research is that many researches are for a specific application domain and are lacking in generality, and generally it is hard to apply the framework of the existing research to other fields. Moreover, it is thought that practical usage of agent is in immature state that there are few practical application examples and the system development methodology is not sufficient. In this research, the cooperative problem solving system by the multi-agent is first shown as an application development framework by the agents. Then, we discuss usage of templates of the agent model and the task model as the development support technique and the evaluation according to an application construction experiment in a planning problem or the field of network community support is considered.

1. はじめに

コンピュータやデジタルネットワークの利用が広がり産業活動や生活の中に浸透してくるにつれ、高度なネットワークアプリケーションの実現が期待されている。企業などの組織内の知識を連携・協調させて柔軟な問題解決が行えるようなしくみや、ネットワーク上での人々の社会活動（協調、交渉、連携、仲介）を代行・支援し、ネットワーク社会での快適で効率的な生活を提供するしくみといったものが求められている。

エージェントは、社会的な営みをネットワーク上で実現することが可能な自律性、社会性等をもったソフトウェアシステムとして研究開発成果が期待されている。また、エージェント間のメッセージ通信機能等プラットフォームレベルでの国際標準化の活動が進められている。エージェントシステムが互いに協力し合い、

利用者の為に様々な情報検索や情報処理作業を代行し、利用者の利便性を高める「エージェント社会」の実現が近いと言われている。

しかしながら、エージェント技術の現状での研究は、特定の応用領域を対象にした研究が多く行われており、得られた結果が一般性に乏しく、既存の研究の枠組みが一般的には他の分野には適用しにくいといったレベルに留まっている [1, 2]。また、実用面でもシステムの開発方式の整備が不十分、実用問題への適用事例が少ないといった未成熟な状況であると考えられる。

これは、アプリケーション構築において、エージェントシステム構築の方法論が確立されるに至っておらず、応用毎にエージェントモデルや処理系をアドホックに選択しているといった状況がある。また、実用的な評価事例に乏しく、エージェント応用システムの生産性、性能といった評価が十分得られていないといったことが要因と考えられる [3]。

このような課題に対応して、アプリケーション構築の為にエージェントアーキテクチャとしてエージェン

[†] 三菱電機 (株) Mitsubishi Electric Corp.

^{††} 東海大学, Tokai University

^{†††} 東京電機大学, Tokyo Denki University

トモデルや構築方法を開発し、実用レベルの評価を行いながら、アプリケーションの種類に対応した共通の構築基盤として整備していくことが求められている [4, 5, 6]

本研究では、こういった背景を踏まえ、マルチエージェントによる協調問題解決を対象として、まずエージェントによるアプリケーション構築基盤である、マルチエージェントによる問題解決方式について論じる。さらに、開発支援手法としてエージェントモデルとタスクモデルによるテンプレートを構築し利用する方式について提案する。

2. マルチエージェントによる協調問題解決方式

エージェントは、自律性、協調性、適応能力、代理性、移動性といった性質を持つソフトウェアシステムとして定義され研究されている [1]。FIPA [7] では、エージェント管理、ディレクトリ、エージェント通信路等のプラットフォームレベルの機能定義が進められている。

本研究で提案するエージェントの基本モデルを図1に示す。知識による意思決定機能により自律性を実現する。他エージェントを含む外部環境との協調・交渉・適用等により処理を行う。意思決定機構は、内部構造を、信念 (B: Belief)、欲求 (D: Desire)、意図 (I: Intention) という BDI モデルによって考える [8]。Belief (信念、現在自分を取り巻いている状況に関する情報)、Desire (達成しようとしている願望)、Intention (実行目標) という内部構造により、エージェントが自らのおかれた状況を常に監視して目標を達成すべくプランニングを実行し行動するという枠組である。

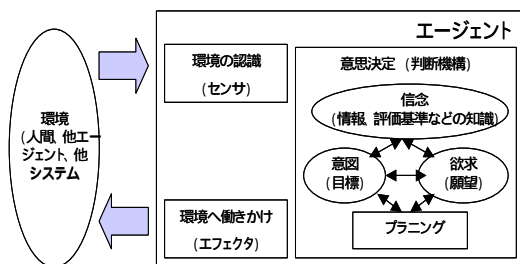


図1 エージェントの基本モデル

2.1 基本モデル構成

本研究で提案するエージェントアーキテクチャの基本モデル構成を、FIPA が提案するプラットフォーム上に位置づけたものを図2に示す。本研究では、上位のエージェント基本機能として、エージェント通信機能、知識の共有化形式、意思決定・判断機構を共通に持つ方式とする。意思決定機構は、内部構造として信念 (状況に関する情報、知識)、意図 (実行目標) 等を持っている。

問題の解決方式として、トップダウン的な分割・統

合型エージェント方式とボトムアップ的な分散・連邦型エージェント方式およびそれらの組合せ方式を考える。図2に示すように、方式によって備えるべき機能が異なってくる。前者では、事前に問題が与えられ、部分問題に分割された後、その部分問題に対して担当エージェントを設計する方式を取る。後者では、エージェントへの問題割当ては動的に行われ、割当てを受けるかどうかは自律的なエージェントの判断による。ここでは、交渉や仲介の機能が重要となる。

本研究では、共通的な機能の抽出を行い、エージェント基本機能として、エージェント通信機能、知識の共有化機能、意思決定・判断機構を共通基盤ライブラリとして持つ方式とした。これにより、トップダウン的な方式とボトムアップ的な方式を容易に組合せながら柔軟な問題解決が可能な枠組みの基盤を与えることができる。

例えば、計画問題を取り上げた場合、問題解決のフェーズにより、トップダウン的な方式とボトムアップ的な方式を連続的に適用することが有効と考えられる。

| | |
|-------|------------|
| 初期計画 | トップダウン的な方式 |
| 再計画 | トップダウン的な方式 |
| 動的な計画 | ボトムアップ的な方式 |

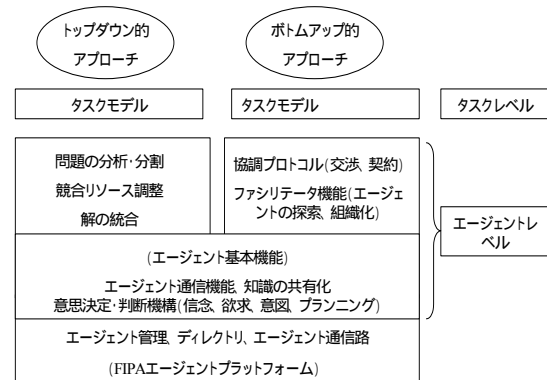


図2 エージェントによる問題解決の基本モデル

2.2 方式の検討

本研究では、以下に示すように、分割・統合型エージェントと分散・連邦型エージェントによる問題解決方式について検討を行う。

2.2.1 分割・統合型エージェント方式

大規模な設計、計画等の業務において問題解決を効率的に実行するコンピュータ支援が求められている。エージェント技術による研究も種々進められているが、複数の異なるレベルの知識を用いた問題解決モデルの設定、実用的な時間で制約を満たした許容解を導出する方式の確立、といった課題は未解決である。この課題に対し、複数の人間による協調的な問題解決をモデルとし、部分問題への分割と解の統合を特徴とする分割・統合型エージェント方式が有効と考えられる。

本方式では、与えられた問題は、分析・分割・整理により、いくつかの部分問題に分割される。そして、各部分問題を担当するいわゆるエージェント群をトップダウンに設計し、それらのエージェントを組み合わせ

せてエージェントシステムを実現する。

図3に示すように、基本的なエージェント群は、複数の部分問題を解決する部分問題エージェント、全体解を統合する統合エージェント、各種の与えられた条件や制約等の管理を行う管理エージェントから構成される。

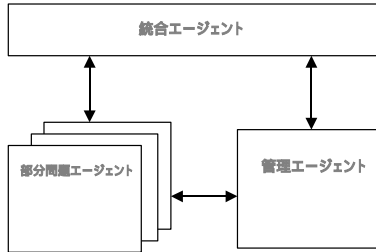


図3 分割・統合型エージェント方式の構成

- 統合エージェント
部分問題への分割と全体解を統合
- 部分問題エージェント
各種の制約知識を制約ネットワークとして保持し、制約下での組合せ問題を解決
- 管理エージェント
業務条件等の知識を保持し管理
エージェント間は、「通知」、「依頼」、「拒否」、「問合せ」といったエージェント通信手段に基づく協調動作を行う。

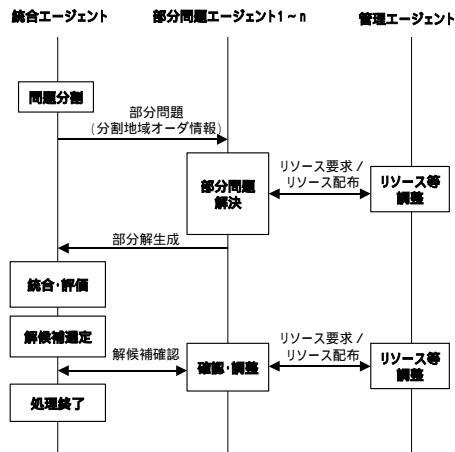


図4 マルチエージェント間プロトコル

マルチエージェント間プロトコルを図4に示す。また、プロトコルの詳細を以下に示す。

統合エージェントによる問題分割と部分問題エージェントへの部分問題の依頼
部分問題エージェントによる部分問題解の生成と、統合エージェントへの通知（各部分内での制約を満たす組合せの生成）
部分問題エージェントでの、問題解決に必要なリソースの管理エージェントへの依頼と管理エー

ントによるリソースの割当てと通知
統合エージェントによる、部分問題解の統合による許容解の生成
許容解の選択（ユーザによる選択も含め）と、部分問題エージェントへの通知
部分問題エージェントにおける、最終的なリソース等の調整
最終的な解の出力

(1) エージェントにおける知識と問題解決

図5に示すように、統合エージェントでは、部分問題への分割と部分解の統合知識を割り当てる。部分問題エージェントでは、各種の制約知識を制約ネットワークとして保持し、制約下での組合せ問題を解決する。管理エージェントでは、業務条件等の知識を保持し管理を行う。

(2) エージェント間通信と協調

エージェント間は、「通知」、「依頼」、「拒否」、「問合せ」といったエージェント通信(ACL)に基づくメッセージ通信によって情報交換や協調動作を行っている。分割・統合型では、各エージェントへのタスクの割当ては静的であり、協調はメッセージの交換レベルで行われる。

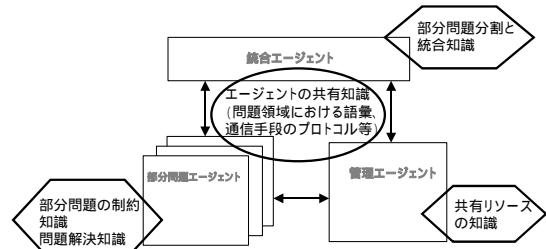


図5 マルチエージェント間の知識の共有

2.2.2 分散・連邦型エージェント方式

分散・連邦型エージェントは、自律性の強い多数のエージェントが互いに協力し合い、ユーザのために様々な情報処理作業を代行していく環境を実現する。オープンなネットワーク社会をベースにしたエージェント社会の実現には、既に存在しているエージェントの集団を組織化し問題解決に当たる事が重要になってくる。本研究では、エージェント集団へのリアルタイムでのタスク割当てを特徴とする分散・連邦型エージェント方式を検討する。図6、7に示すように、分割・統合型との中間的な形態の階層タイプと、フラットにエージェント群が集団を形成する分散タイプがあると考えるがここでは分散タイプを主に考える。

(1) 階層タイプ

階層タイプでは、動的なタスクの割当てを協調的に行う為、統合エージェント・各管理エージェント間および管理エージェント・各エージェント間での契約ネットワークプロトコルを導入する。各エージェント群は、管理エージェントをファシリテータとする連邦アーキテ

クチャと見ることもできる。

問題解決では、分割・統合型と同様に問題分割に伴う解の最適性と収束性におけるトレードオフを考える必要がある。

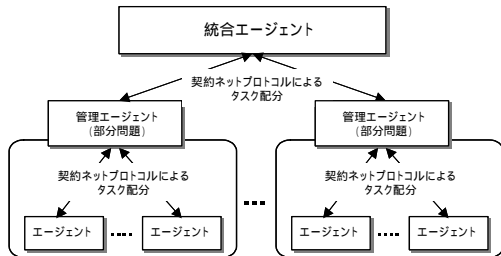


図6 分散・連邦型エージェント方式（階層タイプ）の構成

(2) 分散タイプ

エージェントは、情報と内部状態に応じてその時点での目標を決め、自律的にプランを生成して問題解決を行っていく。エージェントは、通常はネットワーク上の集団として構成され連邦アーキテクチャの形をとる。集団を管理するエージェントが存在し、その働きによって緩く結合したコミュニティ的なものから、強い目的志向の集団まで特性が異なってくる。

マルチエージェント間プロトコルを図8に示す。また、プロトコルの詳細を以下に示す

分散・連邦型マルチエージェント方式においては、例えば、契約ネットプロトコルと呼ばれる、タスクの公開通知、応札、タスクの受託・実施、といった協調に基づくプロトコルを利用する。ある集団から他の集団群へのタスクの実行依頼は以下の手順で行われる。

他の管理エージェント（タスクの依頼者）から、ある集団の管理エージェント（複数）にタスクを公示

管理エージェントは、必要であればタスクの分割等を行い、自集団内のエージェントにタスクを公示し、エージェントから応札を求める

公示を受信したエージェントは、コスト見積もりを行い応札する（応札を拒否してもよい）

管理エージェントは、受け取った応札を評価し、まとめることによって集団としての応札を行なう（応札を拒否してもよい）

タスクの依頼者である管理エージェントは、受け取った応札を評価しタスクを発注する集団を決定し、発注を行なう

発注を受け取った管理エージェントは、集団内のエージェントにタスク発注を行なう

発注を受けたエージェントは、受諾メッセージを送る

集団の管理者エージェントは、タスクの依頼者である管理エージェントに受諾メッセージを送る

マルチエージェント毎の個別知識と、マルチエージェント間の共有知識の様子を図9に示す。共有知識としては、全体社会のルール、問題領域における語彙、

通信手段のプロトコル等、各エージェントが持つことにより協調動作が支障なく行われる。

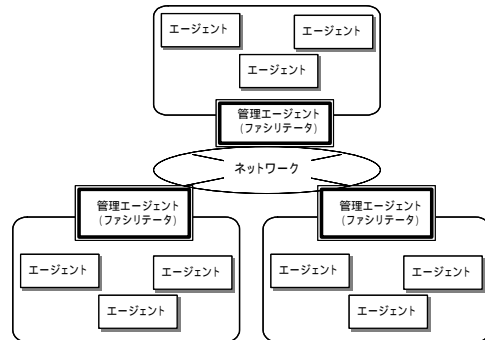


図7 分散・連邦型エージェント方式（分散タイプ）の構成

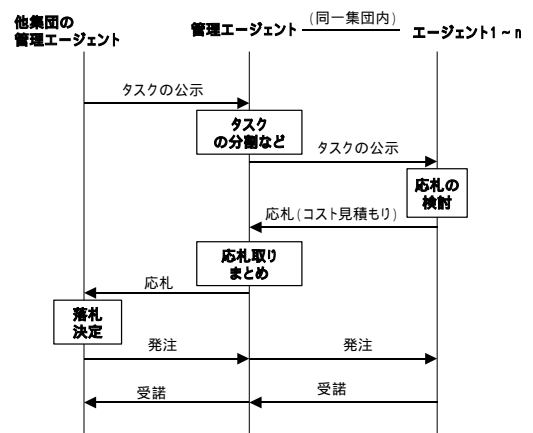


図8 マルチエージェント間プロトコル

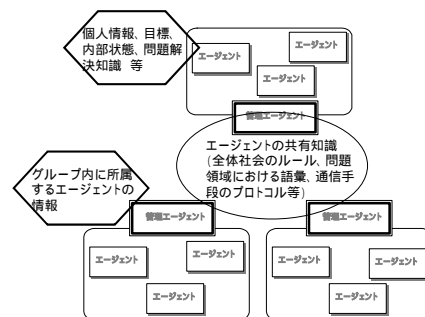


図9 マルチエージェント間の知識共有

3. 開発支援手法

ここで提案する開発支援手法は、分析レベルから実装レベルまでの開発支援手法の確立を目指すものである。2章で述べたエージェント方式に対応したエージェントモデルとタスクモデルについて、まず述べ、それらを統合したエージェントシステム構築の手順について述べる。

3.1 エージェントモデルとテンプレート

エージェント方式に基づくエージェントテンプレ

トの構築について考える。

タスクに依存しないもので、モジュール構造のフレームワーク、タスク実行の推論エンジン、外部とのインタラクション機能等をエージェントテンプレートとして設定する。

(1) 分割・統合型エージェント

分散・統合型エージェントの枠組みを適用したエージェントモデル階層の例を図10に示す。各ノードに対応してエージェントテンプレートが設定される。

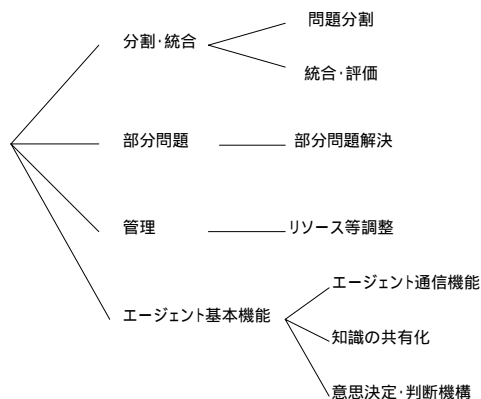


図10 エージェントモデルの階層（分割・統合）

(2) 分散・連邦型エージェントにおけるエージェントモデル

分散・連邦型階層タイプの枠組みを適用したエージェントモデル階層を図11に示す。

階層タイプによる計画実行システムにおけるマルチエージェント意思決定支援、分散タイプによるコミュニティ形成支援の構築等への適用を想定する。

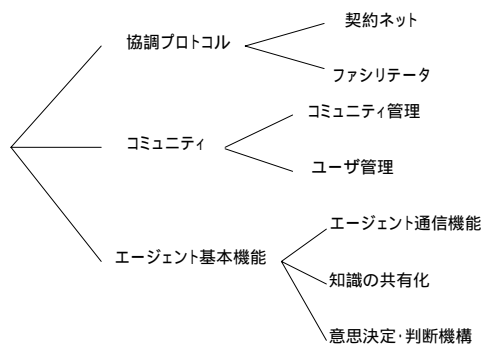


図11 エージェントモデルの階層（分散・連邦）

協調プロトコルとしては、契約ネットプロトコルやファシリテータによる仲介プロトコルによる協調パターンを代表的なものとして取り上げる。協調プロトコルの適用方法により、階層的な作業分担による再計画処理、時間制約付きの再計画処理、といったことが可能となる柔軟な枠組みの実現が可能となる。

3.2 タスクモデルとテンプレート

タスクは、汎化タスク（Generic Tasks）、タスクオントロジーなどの研究により種々パターン化が提案されている状況である。ただ、汎化タスクは、粒度が大きくてさまざまな形態を持つ現実のタスクを表現することができないなどの問題がある。新たな手法としてタスクオントロジーの手法が提案されている[6]。本研究では、エージェントシステム構築におけるタスクレベルのデザインパターンの利用を考え、仕様レベルの抽象記述と、実装レベルの詳細記述を統合的にサポートする方式を提案する。再利用可能なライブラリを準備することにより、タスクレベルの抽象度を保ちつつ実装レベルの開発支援環境を提供することを考える。

(1) タスクオントロジーと知識構造

複雑で大規模な問題を解決するために、問題をサブ問題へと階層的に構造化することを考える。

ここにおける、タスクオントロジーは、問題解決のための知識を抽出し整理したものである。スケジューリング問題におけるタスクオントロジーの例を図12に示す。

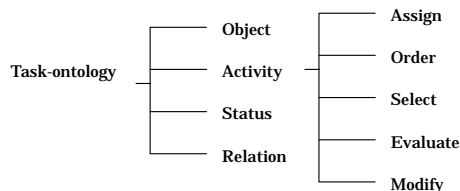


図12 タスクオントロジーの例

タスクオントロジーに基づく汎用的な問題解決構造を考える。問題解決構造は階層的な構造で表現できる。各ノードは問題解決エージェントで下位の階層ノードはより上位のノードのサブ問題を解決する。図13にスケジューリング問題の例を示す。具体的な問題解決構造はジョブのラインへの割付、ライン内のジョブの立案順序決定、ジョブ内の工程決定、解の評価、修正の各ノードから成る。

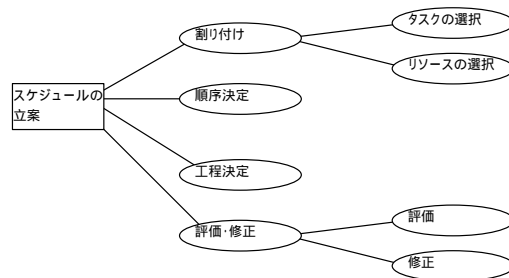


図13 タスクオントロジーに基づく問題解決構造例

ここで、ノードタイプのパターンを整理し、デザインパターンとして再利用することを考える。スケジューリングの為のタスクオントロジーから抽出されたノードタイプのパターン例を表1に示す。これらのノードを階層的に構成することによりスケジューリング問題に対する問題解決構造を容易に作成可能となる。

表1 ノードタイプのパターン例

| ノードタイプ | 機能の説明 |
|-----------------|---------------------------|
| Assign R, T | タスク T にリソース R を割り付ける |
| Order T1,..,Tn | タスク T1 から Tn を順序付ける |
| Select S, A, C | ソース S から制約 C を満たすもの A を選ぶ |
| Evaluate S T, O | 状態 S T を評価し結果を O へ返す |
| Modify C | スケジュールを修正する |

(2) デザインパターンの実装

ここでは、知識システム構築環境であるスキーマシステムによるノードタイプの実装例を図14に示す。ノードタイプはスキーマによる知識表現を取ることにより再利用を考慮した実装とすることが可能である。

```

schema assign subclass_of: design_pattern {
  R{type resource}
  T{type task}
  assign = method(?rsrc, ?tsk){}

Select S, A, C
schema select subclass_of: design_pattern {
  S{type source}
  A{type result}
  C{type constraint}
  select = method(?src, ?rslt, ?cstrt){}

```

図14 ノードタイプの実装例

3.3 エージェントとタスクの統合

タスクテンプレートの選択とエージェントテンプレートの選択によりシステムを構築していくことになるが、エージェント方式によりその手順は異なってくる。以下に其々の手順について述べる。

- (1) トップダウン的な分割・統合方式
 - ・ 対象ドメインおよびタスクの分析
 - ・ タスクテンプレートの選択とタスクモデルの詳細化
 - ・ タスクをエージェントに割振ることによるエージェント機能の定式化とエージェントテンプレートの選択およびエージェントモデルの詳細化
- (2) ボトムアップ的な分散・連邦方式
 - ・ 対象ドメインの分析
 - ・ マクロなエージェント構成(ファシリテータ、交渉モデル、コミュニティモデルなど)をエージェントテンプレートの選択により詳細化
 - ・ 対象タスクの分析
 - ・ タスクテンプレートの選択とタスクモデルの詳細化
 - ・ マクロなエージェント構成に応じたタスクのエージェントへの割振り
 - ・ エージェントモデルの詳細化

エージェントモデルとタスクモデルを階層的に配置し、それらを統合していくことによりシステムを構築していくことを提案した。トップダウン的には、タスクをエージェントに割振ることになる。ボトムアップ的に

は、予め用意されたエージェント構成においてエージェント自らが与えられたタスクを交渉等により配分することにより問題解決を行っていく。

4. 提案手法の事例への適用と評価

提案手法を計画問題やネットワークコミュニティ支援の分野での応用構築に適用し、評価を実施する。

4.1 分割・統合型エージェントによる配送計画支援システムの構築

配送計画が解決しようとする問題は、複数のトラックが配送センターから多数の届先に荷物を届ける場合、乗務員、配車の割当、配送ルート選択等の計画を配送コストを考慮しながら立案する。

配送対象地域を幾つかの配送区域に分割して考えることにより配送区域毎の問題解決エージェントと全体解の統合・評価を行うエージェント、乗務員・車両といったリソース管理を行うエージェントから構成される分割・統合型エージェント方式によるエージェント構成を図15に示す。

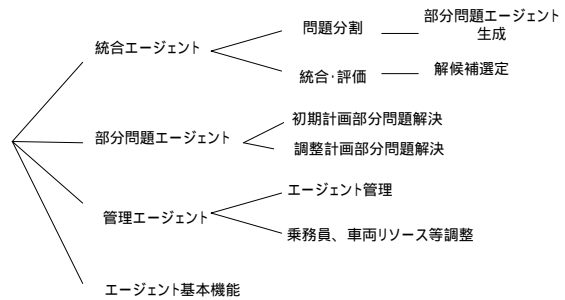


図15 配送計画のエージェント構成

評価・検証として、以下の適用を行った。

(1) 検証事例

ここで取り上げる配送計画問題は、1つの配送センターから約120の届先へ、注文を配送する。配送地域の配送区域への分割数4、トラック数15台、乗務員数10~15という規模の問題を考える。アウトプットとして、配送ルート設定、乗務員・車両の割当て、センター・届先の出発時刻、到着予定時刻を決定する。評価パラメータは、乗務員と車両のコスト、乗務員の作業量の平均度、配送期限に対する余裕度とした。

(2) 評価と考察

検証システムでは、配送オーダー、乗務員、車両、道路情報等に関する実用レベルでの属性情報や各種制約条件を考慮し、計画を立案することができた。本枠組みを用いることにより、見通しのよい配送計画システムが構成できた。例えば、初期計画と調整計画の2段階に計画を分けることも、レベルの異なるエージェント間の調整という形で自然に実現できた。検証システムの規模の問題では、組合せ的爆発も発生しなかった。これは、分散協調による問題の分割と大域的バックトラックの回避による効果と考える。

(3) 解の最適性と収束性

分割・統合型では、与えられた問題を部分問題に分割し、配分を行う。分割の基準は構造的や意味的など問題によって設定される。分割することによって、部分問題の解法は計算コスト的に抑えることができ、解の収束性は良くなる。ここで重要なのは部分問題への分割が必ずしも独立でない場合、解の干渉により統合解は最適解への近似解となる。よって、分割・統合型では、問題解決において解の最適性と収束性におけるトレードオフを考えていくことが重要と考えられる。

4.2 分散・連邦型エージェントによるコミュニティ形成支援システムの構築（分散タイプ）

ネットワーク社会において趣味などのサークル集団の形成をマルチエージェントによりモデル化し、ネットワークコミュニティの形成を支援するシステムを構築する。マルチエージェントを用いてサークル集団の形成をおこない、利用者にコミュニケーションの場を提供することにより、自分の欲しい情報の容易な取得などを支援する。

ユーザは自分の代理人となるエージェントに自分の個人情報を主観情報としてあたえる。またユーザとエージェントは相互にメッセージのやり取りを行う。エージェントは、他のユーザの代理人であるエージェントと主観情報などをもとに相互作用し、コミュニティの形成を行う。コミュニティのエージェントモデルを図16に示す。

コミュニティに参加するユーザを代理する。ユーザの個人情報を管理し、ユーザの代行でパーソナルエージェント(PA)間、PAとコミュニティエージェント(CA)間の相互の情報交換、交渉を行う。同時にユーザとのやり取りを行いながらユーザの最大満足度を追求する。

ユーザの個人情報を反映したコミュニティの形成方式により評価実験を行い、エージェント方式の有効性を確認した。エージェントの内部パラメータによるコミュニティの形成への影響などをさらに評価していく必要がある。

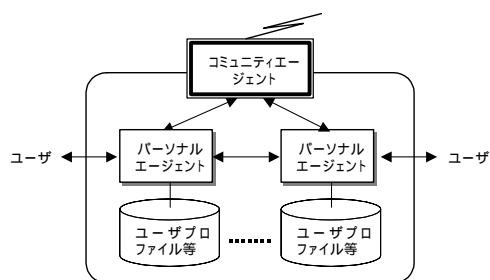


図16 コミュニティのエージェントモデル

4.3 提案手法の評価・議論

エージェントによる問題解決の基本モデルを考えることにより、問題解決方式におけるエージェントの役割・機能を明確化した。分割・統合型方式ではエージェントのオブジェクト的機能が有効であり、分散・連邦型方式ではエージェントの分散・自律的機能が有効であると思われる。このように、エージェントには広範囲の役割・機能を持たせることができる。問題の設定により問題解決方式を選択可能とすることより、エ

ージェントによる目的システムの構築を容易にすることができると考えられる。

エージェント通信機能、知識の共有化機能、意思決定・判断機能を、エージェント基本機能として提供する方式とした。この方式は、問題解決方式毎の機能開発に有効であり、また、基本機能をエージェントに共通的に持たせることにより、異なる問題解決方式におけるエージェントを組合せてシステムを構築することが可能であると考えられる。

本研究では、提案したエージェント方式を応用システムの構築に適用評価することにより、設計、計画等の組合せ問題および動的なタスク配分問題に対する有効性を確認した。

5. まとめ

本研究では、エージェントによるアプリケーション構築基盤として、分割・統合型と分散・連邦型の2種類のマルチエージェントによる協調問題解決方式を提案した。これは、従来から行われているトップダウン的なシステム構築アプローチと、自律的なエージェントモデルを活用したボトムアップ的な構築アプローチをそれぞれ問題解決方式として整理したものである。問題解決方式に対応したシステム開発支援手法を考えることによりよりきめ細かい効率的なエージェントシステム開発支援が行えると考えた。開発支援手法としてエージェントモデルとタスクモデルによるテンプレート構築し利用する方式について検討しその統合方式について述べた。配送計画支援問題やネットワークコミュニティ支援の分野での応用構築実験により、方式の評価を行い有効性を確認した。今後、さらに評価実験を行うと併に手法の強化や開発支援環境の整備を行っていく予定である。

参考文献

- (1) Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. Knowledge Engineering Review, Vol. 10(2), 1995.
- (2) P. Maes, "Agents that reduce work and information overload," Communication of ACM, vol.37, No.7, pp.30-40, 1994.
- (3) N. Jennings, "On agent-based software engineering," Artif. Intell., vol.117, no.2, pp.277-296, Elsevier, 2000.
- (4) Y. Adior, D.B. Lange, "Agent design patterns: Element of agent application design," Proc. Autonomous Agents'98, pp.108-115, 1998.
- (5) 和泉 憲明, 山口 高平, "オントロジーに基づくソフトウェアエージェントのパターン指向開発," 信学論 Vol.J84-D1 No.8, pp.1181-1190, 2001.
- (6) 瀬田 和久 池田 満 島 輝行 角所 収 溝口 理一郎 問題解決オントロジーに基づく概念レベルプログラミング環境 CLEPE 信学論 Vol.J81-D2 No.9 pp.2168-2180, 1998.
- (7) FIPA(Foundation for Intelligent Physical Agents): <http://www.fipa.org/>.
- (8) Anand S. Rao, Michael P. Georgeff, Modeling Rational Agents within a BDI-Architecture, Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91) 1991.