JavaSpacesを使用したフィジカルエージェントの統合*

4R－5

大林真人†　西山裕之†　溝口文雄†

東京理科大学 理工学部‡

1　はじめに

複数のロボット等の物理エージェントを統合し，協調させて運用する結果，1台のロボットでは，遂行することのできない複雑なタスクをも実現することが可能となる．この統合には，ロボット間の競合の解消，契約ネットプロトコル[3]に基づくタスクの交渉等のシステムを必要とする．また，複数の物理エージェント間による協調タスクにおいては，個々のエージェントのタスクの同期処理の実現を必要とする．

本研究においては，Javaによる分散コンピューティング技術を使用し，物理エージェントのタスクやエージェント間における交渉等をオブジェクトとして扱い，JavaSpacesによるオブジェクトの交換を利用して，分散制御された複数の物理エージェントの統合をおこなう手法を提案し，その有効性を検討する．

2　設計方針

従来，複数の物理エージェントを統合する手法として，MRL[2]のようなマルチエージェント用言語を用いる手法や，FIPA[1]のような仕様を用いる方法がある．MRLは並列型論理言語を基にして記述された言語であり，その並列動作の特徴を使用して，各エージェントを並列に動作させ，エージェント間の交渉，協調をサポートする．しかし，MRLによるシステムは特定のホストでのみ動作するため，負荷がそのホストに集中し，その結果，動作できるエージェントの数も限られる．また，新たな物理エージェントの追加，削除に対しては，システムの停止，ソースコードの修正と再コンパイルを必要とする．一方，FIPAにおいては，分散環境におけるエージェント管理や通信言語についての優れた仕様が存在するが，マルチロボットの運用における競合の解消や，協調によるタスクレベルコントロールをサポートしてはいない．

本研究においては，分散制御された，ロボットに代表される物理エージェントを統合するシステムをJava分散コンピューティングを使用して実装する．本システムの主な目的を次に示す．

- 分散コンピューティング技術でシステムを構築することによりエージェントを分散させる．
- 物理エージェントの追加，削除を動的にする．
- タスクをオブジェクトとして扱い，エージェント間の協調タスクの生成を容易に実現する．

3　実装

3.1　JavaSpacesについて

JavaSpacesとは，オブジェクトの交換を仲介する，Javaを使用した分散オブジェクトシステムである．各エージェントは，あるサーバー上に存在するSpaceに対してエントリ（オブジェクト）を書き込んだり，それを取り出したりすることで，他のエージェントとのオブジェクトの交換を実現する．JavaSpacesに対する操作として，Space内のオブジェクトの排他処理が可能であるため，これにより，エージェント間における競合解消や協調を容易に実現することが可能である．本システムにおいては，物理エージェント同士の競合解消，および契約ネットプロトコルに基づくタスクの制御をJavaSpacesを使用したオブジェクトの受渡しによって実現する．

3.2　システム概要

図1に本システムにおけるシステムの構成図を示す．システムの最小構成単位はプラットホームと呼ばれ，物理エージェントの統合をおこなうためのいくつかのシステム部分を含んでいる．

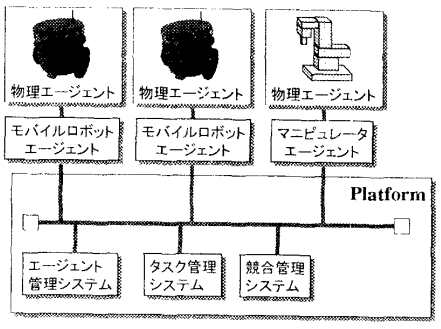

図1: プラットホーム構造

*The integration method for distributed physical agents using JavaSpaces

†Makoto OBAYASHI, Hiroyuki NISHIYAMA, Fumio MIZOGUCHI

‡Dept.of Industrial Admin.Faculty of Sci.and Tech. Science University of Tokyo

3.3 協調におけるタスク管理

3.3.1 タスクの依頼と受託

エージェントによるタスクの依頼，受託，委託の実装は，JavaSpaces 内におけるエントリに対する操作で行なわれる．そのシーケンスを以下に示す．

1. 依頼を行なうエージェントは，依頼の内容と委託の対象となるエージェントを示すエントリを Space 内に書き込む．
2. 依頼を受けるエージェントは，そのエントリを Space から取りだし，受託の意思と依頼されたタスクの遂行に対するコストを記述する．
3. 依頼を行なったエージェントは，一定時間後にオブジェクトを Space から回収し，コストの最も低いエージェントに対して依頼の委託を行なう．

3.3.2 タスクのオブジェクト化とその利点

協調タスクの例題として，マニピュレータが移動ロボットにプリンタから印刷された用紙を渡すタスクを考える．このとき，このタスクは次の 3 つの細かいタスクの集合であると考えられる．

1) 移動ロボットが受渡し位置まで移動する．
2) プリンタが用紙の印刷処理を行なう．
3) マニピュレータが移動ロボットに印刷された用紙を渡す．

本研究におけるシステムでは，1 つの協調タスクを上記のようなサブタスクに分解し，それぞれをオブジェクトとして取り扱う．それらは，タスクの種類を表す type，オブジェクトを一意に識別する ID，タスクの成功，失敗を示す result 等のフィールドをもつ．

この様に，物理エージェントのサブタスクをオブジェクトとして構成することにより，複数のサブタスクから，最終的な目標を含む一つのタスクを，状況にしたがって動的に生成することが容易となる．また，複数のロボットによる協調を必要とするタスクにおいては，各ロボットが，サブタスクの終了時にそのオブジェクトを Space を介して他のエージェントに渡すことにより，同期処理を容易に実現することが可能となる．

3.4 物理エージェントの競合解消

物理エージェント間における競合の解消もまた，各エージェントによる JavaSpaces を介したオブジェクトの参照によって容易に実現できる．移動ロボット同士における競合の解消については，環境内におけるノード情報，移動経路の情報とを保持するオブジェクトを管理し，移動ロボットを制御する各エージェントが，そのオブジェクトを排他的に参照，操作することによって競合を回避する．

4 評価

次に示すのは，本システムを実際の環境において，例題タスクを行なった実験例である．例題タスクには，”プリントデリバリ” を用いた．このタスクはユーザからの依頼によって，移動ロボットがプリンタから印刷された用紙をユーザの居場所まで運搬するものである．印刷された用紙を移動ロボットに渡すタスクは，マニピュレータが行なう．このタスクには，移動ロボット，マニピュレータ，プリンタの 3 つの物理エージェントと，ユーザとの依頼の仲介を行なうインターフェースエージェントとの協調によるタスクの遂行が必要となる．図 2は，エージェント間における通信のタイムチャートである．

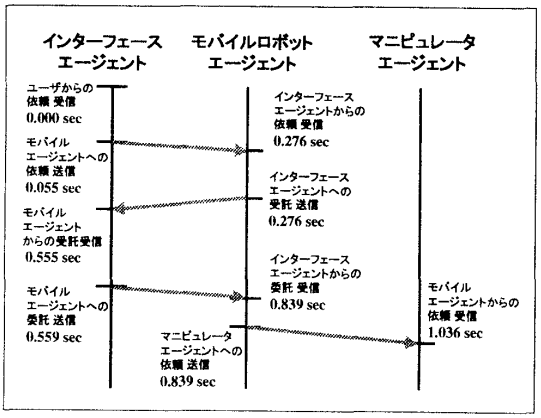

図 2: エージェント間通信におけるタイムチャート

図よりエージェント間におけるタスクの同期，および交渉が確実に達成されていることが確認できる．エージェント間における交渉の速度はやや遅いがと思われるが，例題のタスクに対しては問題にならない速度である．

5 結論

本研究では，Java 分散コンピューティング技術を使用して，複数の物理エージェントを統合する手法を提案し，その実用性を示した．本手法においては，各エージェントのタスク，エージェント間の交渉をオブジェクトとして扱い，タスクの同期処理，エージェント間の競合の解消等を実現し，統合をおこなった．

参考文献

[1] FIPA Specification, *http://www.cselt.it/fipa/*

[2] Hiroyuki Nishiyama, Hayato Ohwada and Fumio Mizoguchi, A Multiagent Robot Language for Communication and Concurrenc y Control, *International Conference on Multiagent Systems(ICMAS)*, 1998.

[3] R.Smith, The Contract Net Protocol:High-Level Communication and Control in a Distributed Problem Solver, *IEEE Transaction on Computers*, Vol.C-29,No.12,1980.