メディエータを用いた物理エージェントとソフトウェアエージェントの協調環境の構築

岩田 浩司　鷹合 基行　沢田 篤史　武田 英明　西田 豊明

奈良先端科学技術大学院大学 情報科学研究科

〒 630-01 奈良県生駒市高山町 8916-5

tel: 07437-9-9211 (内線 **5316**)

{**kouji-i,motoyu-t,sawada,takeda,nishida**}**@is.aist-nara.ac.jp**

人間とロボットと計算機が実世界において互いに協調できる環境をマルチエージェント系で実現するのは一つの有効な方法である。本稿では、異なる機能を有するエージェント間の協調に、仲介を行なう特別なエージェントであるメディエータを用いる方法を提案する。メディエータは、共通オントロジーによる領域知識管理機能、タスク分解機能、エージェント能力管理機能、エージェント知識収集機能を有し、エージェント固有の知識とエージェント間で共有可能な知識との変換を行なう。ここでは、自律型棚、自律型ドア、搬送行動型エージェント、アームを有する行動型エージェントなどの物理エージェント、および計算機上のソフトウェアエージェントを統合し、協調作業支援環境を構築する際に作成したメディエータについて述べる。

キーワード　　マルチエージェント, ロボット, 仲介, 協調

A Cooperative Environment for Real-world Agents and Information Agents with Mediation

Kouji IWATA　Motoyuki TAKAAI　Atsushi SAWADA　Hideaki TAKEDA　Toyoaki NISHIDA

Graduate School of Information Science,

Nara Institute of Science and Technology,

8916-5, Takayama, Ikoma, Nara 630-01, JAPAN

tel: +81-7437-9-9211 (ext. 5316)

{**kouji-i,motoyu-t,sawada,takeda,nishida**}**@is.aist-nara.ac.jp**

We build the cooperative environment composed of human,robots and computers based on a multi-agent architecture.In this paper,we focus on a special agent,called mediator ,that facilitates communication between those heterogeneous agents. To cooperate each other by this communication,mediator should support the function to maintain the knowledge about the domain with the common ontology ,the function to decompose the task ,the function to manage the agents' ability ,and the function to collect the knowledge dependent on each agent. To illustrate the mediator ,we show a system which consists of real-world agents for autonomous mobile robots,and intelligent environments like shelf and door,and information agents for imageprocessing and so on. The mediator interprets requests from human,and successfully decomposes and distributes tasks to agents according to agents' ability and their situations.

keywords　　Multi-agent, Robot, Mediator, Cooperation

1 はじめに

我々は、人間とロボットおよび計算機上のソフトウェアが協調する環境を構築することで、人間がより知的な作業を行える環境の構築を目指している。

この支援環境を構築する際には、単体のロボットで実現するヒューマノイド作成によるアプローチと、複数台のロボットや計算機、また時には人間が互いに協力する協調によるアプローチがある。後者のアプローチをとることにより、機能が分散されハードウェア作成が容易になる。また、機能の組み合わせを可変にすることができるため、より柔軟なシステムが構築できる。一般に、人間の支援を行なうシステムには柔軟な対応が求められるので、我々は異なる機能を有するロボット、および計算機、さらに人間が互いに協調する後者のアプローチが有効であると考える。

この協調を実現するには、ロボットおよび計算機上のソフトウェア、また人間の間で相互にコミュニケーションを行う自律・協調的な環境の構築が必要である。このコミュニケーションの基盤となるネットワークの構築が重要である。以下、ネットワークの構成要素をエージェントと呼び、ネットワークをエージェントネットワークと呼ぶ。

人間がシステムに加わることで、計算機上では管理しきれない人間の予期せぬ行動も考慮する必要がでてくる。よって、各エージェントが有する機能を用いて、実世界から知識を獲得しシステム内の不完全な知識を補う必要がある。しかし、実世界にあるものがたとえ同一物であっても、異なるエージェントが各々の機能を用いて認識した結果得られる知識は、各エージェントによって、表現形式・表現内容は異なる。本来ロボットは特定の目的をもっていることが多く、知識の表現形式や表現内容は異なっている。

このように、異なるエージェント間での知識の伝達による協調を実現するための機構がエージェントネットワークには必要であり、この機構を持つエージェントをメディエータと呼ぶ。

本稿では、異なる機能を有するエージェントが知識を有し、その知識をエージェント間で伝達することで協調を行う環境において、重要な役割を担う特別なエージェント (メディエータ) の機能とその実装について述べる。

2 協調支援環境

計算機により協調作業を支援する研究分野としては、グループウェアおよび CSCW(Computer Supported Cooperative Work) がある。グループウェアとは共通の仕事や目的のために働く利用者のグループを支援し、共有作業環境のためのインタフェースを提供する計算機ベースのシステムである。一方、 CSCW はグループワークの中でコンピュータの役割に注目する概念や基本姿勢を表したもので、支援 (CS) という支援手段、協同作業 (CW) という支援対象の 2 つの異なる視点を合わせ持った概念である [1]。

実世界でのロボットや計算機上のソフトウェアが協調して人間を支援するシステムを考える時でも、システムの構成要素になるエージェントの役割に注目する。よって、 CSCW と同様に 2 つの異なる視点からみた支援の仕方がある。一つは、エージェントに作業分担させることによる支援であり、他方は人間と協同して作業を行なうエージェントによる支援である。図 1に作業分担による人間の支援の例を記す。また、図 2に協同作業による人間の支援の例を記す。

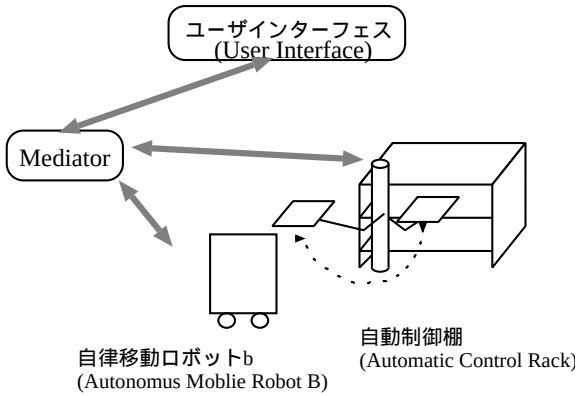

図 1: 作業分担での人間の支援

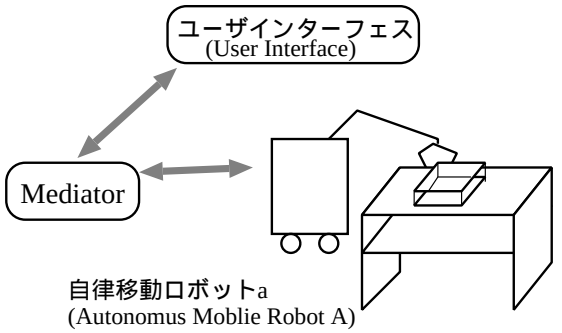

図 2: 協同作業での人間の支援

本研究で想定している人間の作業空間は、ロボットと人間が共存する環境において機器を組み立てるようなワークスペースである。このワークスペースにおいて、作業分担による人間の支援の例としては、組み立て作業中において、必要な工具の取り寄せをエージェントに依頼する支援があげられる。また、協同作業による人間の支援の例としては、組み立て作業中において、そこの端を押さえておいて欲しいと思った時、エージェントが押さえてくれるような支援があげられる。

3 実験環境

実験設備内の物理エージェントには、移動ロボットの知能化を行なった行動型エージェントがある。また、環境側の知能化を行なったエージェントとして、自律型ドアと自律型棚がある。また、ソフトウェアエージェントとして、移動ロボットのカメラからの画像処理を行なう画像処理エージェント、支援システムへのユーザーインタフェースエージェントと統合のためのメディエータを作成した。実験設備について図 3に記す。

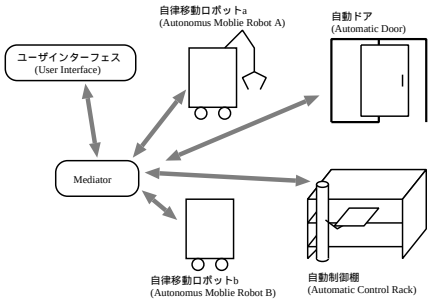

図 3: 実験設備

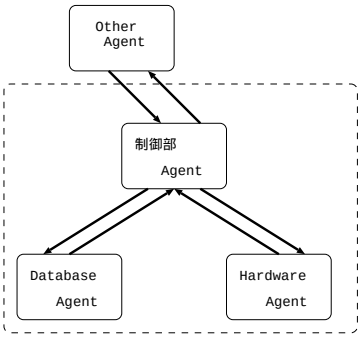

図 4: 物理エージェントの構成

3.1 エージェントネットワーク

エージェントがコミュニケーションを行うために構築されたネットワークをエージェントネットワークと呼ぶ。本研究では、ロボットなどの物理エージェントとメディエータなどのソフトウェアエージェントで構成されている。各エージェント間のコミュニケーションのための共通プロトコルとして、メッセージ伝達行為の表現である KQML(Knowledge Query and Manupulation Language)[3] を使用した。KQML ではメッセージ伝達行為を表現するために、1 対 1 メッセージ伝達を基本として、メッセージの種類を表すキーワード、送り先や送り元などのパラメータ、メッセージ本体からなるフォーマットを規定している。メッセージ内容の表現には命令の表現と同じ表現を用いている。

3.1.1 物理エージェントの構成

本研究では、以下に述べる各機器が自律性を有し、エージェント間通信機能を有する様に図 4に示す様な構成を持つエージェントとして作成した。

- ハードウェアエージェント
  制御エージェントからの走行制御命令などを実行する。また、獲得したセンサー情報などを制御エージェントに伝達する。行動型エージェント kappa1a と kappa1b では、エージェント CPU 上で実行される。
- 制御部エージェント
  他のエージェントからの KQML によるメッセージを解釈して、ハードウェアエージェントが理解できるメッセージに変換する。データベースエージェントと協力して、内部状態の管理をおこなう。
- データベースエージェント
  データベースエージェントは、エージェント自身の状態と獲得した情報の蓄積を行う。

以下にエージェントとして用いた各機器をあげる。

1. 自律型ドア: エージェントネットワーク上のワークステーションからシリアル回線でつながった扉の自動開閉を行うハードウェアの制御を行っている。
2. 自律型棚: エージェントネットワーク上のワークステーションからシリアル回線でつながったハードウェアーの制御を行っている。収めている場所を表す Cell No. を指定することで、棚に収めてある収納物の自動取り出し及び収納を行う。また、搬送テーブルは位置を表す Point No. 指定でも移動可能である。自律型棚の概観を図 5に記す。
3. 行動型エージェント **kappa1a:** 立体視用のカメラを有し、6 自由度でハンドを持つアームを 2 台有する自走型ロボットである。システム全体は、モトローラ社 68040 のボード上で稼働するリアルタイム OS で制御される。以下では、この CPU をエージェント CPU と呼ぶ。概観を図 6に記す。内部に持つ地図データベースに基づき地点間自律移動を行うモードとセンサーベースで移動するモードを有し、互いに切替え可能である。地図データベースは、停止場所である waitpoint と廊下を表す Hall で表されている。エージェント間の通信は、動画像の転送および音声の転送、無線 Ethernet による TCP/IP 通信で行なう。
4. 行動型エージェント **kappa1b:** アームがない点以外 kappa1a と同様の仕様である。搬送を目的としている。
5. **ID** 情報エージェント: 赤外線の検知によりワイヤレス発振機により ID を送出する送信機と、赤外線の発振と、ID の受信を行う受信機から構成されている。

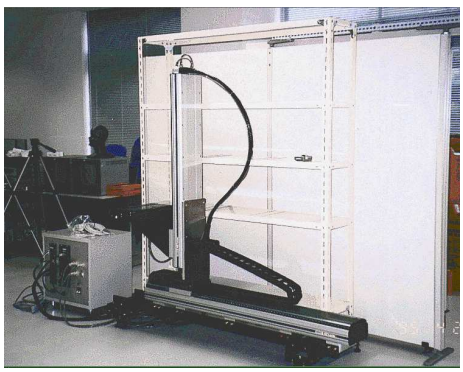

図 5: 自律型棚の概観

3.1.2 ソフトウェアエージェントの構成

1. ユーザインタフェースエージェント: マルチエージェントシステムへの人の命令を入力するため

図 6: kappa1a の概観

のエージェントである。メディエータ及び各エージェントへの命令伝達を行う。
2. 画像処理エージェント: 収納位置の色データベースとの比較による自律型棚エージェントに収めてある場所を見付ける機能と、球状のもの (ラベル) を見付けてその色を読み取り返す機能を有する。
3. メディエータ物理エージェント、ヒューマンインタフェースエージェント間の情報伝達の仲介を行うエージェントである。

4 エージェントに必要な知識

エージェント間で伝達する知識には、エージェント固有の知識とエージェント間で共有可能な知識がある。特に、異なるエージェント間の協調では、エージェント固有の知識をエージェント間で共有可能な知識に変換しなければ互いに知識を伝達できない。また、知識の種類としては、ものに関する知識、タスクに関する知識、空間に関する知識があげられる。

4.1 エージェント固有の知識

- ものに関する知識
  各エージェントが実世界に存在するものの認識を行うにあたり必要な情報である。対象物の RGB 値、重量などが例としてあげられる。
- 空間に関する知識
  各エージェントが移動するために独自に地点にラベル付けした知識である。例としては、自律型棚に収納されている位置を表す Cell No. や kappa1a 及び kappa1b の内部に有する地図データベースの要素である waitpoint などがあげられる。

4.2 エージェント間で共有可能な知識

- ものに関する知識
  各エージェントが実世界に存在する際に共通した情報である。共通知識として領域に関する共通の概念体系 (オントロジー) を用いて表現している。本研究で想定しているワークスペースでの共通オントロジーを図 7に記す。領域に関する共通オントロジーを参照することにより、対象物について不足している情報をそのクラス情報で付加できるようにしている。また、各エージェントにより概念形成能力が異なる場合においても、対象物の上位クラスの伝達による知識の伝達が可能になる。例として、ものの名称、所在地、移動元、移動先、色、重さなどがあげられる。
- 空間に関する知識
  エージェント間の協調動作を行う際においては、特定の場所を互いに認識する必要がある。例えば、ものの手渡しを行なう命令なら互いに手渡しを行う地点を理解しないと実行出来ない。そこで、場所の表現に空間指標を導入し、前置詞や副詞や述語と組み合わせて表現する。空間指標は、エージェント間で共通の領域に関するオントロジー内で表される。エージェント間で互いに理解できる表現を用いることで場所に関する知識を伝達できる。空間指標の例としては、エージェント名および棚の名称などがあげられる。
  以下に場所に関する知識の表現形式について記す。
  1. 前置詞 + 空間指標
     前置詞としては、位相的前置詞である at,on,in、投影的前置詞である in front of, behind, to the right of, to the left of を用いる。
     (例)
     - on(kappa1b)
     - in front of(rack)
  2. 動作に関する述語 + 空間指標
     各命令に依存した知識の表現として用意した。
     (例)
     - grasppoint(desk)
     - releasepoint(desk)
     - viewpoint(rack)
  3. 副詞 + 空間指標
     - here(operator)
  4. エージェント間の協調に必要な地点
     - meetingpoint(kappa1b、 kappa1a)
  5. 状態に関する知識の伝達
     - state(door)
  - タスクに関する知識
    本研究では、命令とその分解方法として記述する。命令は、命令の名称と命令の属性で表現される。

    命令 = 命令名称 (: 属性名 属性値
    : 属性名 属性値
    :
    :
    :)

    命令の属性には、タスクを表すために必要なタスクに関する知識を値とする属性と、エージェントが移動するために必要な移動に関する知識を値とする属性がある。タスクに関する知識が値となる属性名として、運ぶ物の名称が属性値となる object、また運ぶ物の初期地である from_place および最終地である to_place などがあげられる。また、エージェントの移動に関する知識が値となる属性名として、エージェントの現在地を表す from また移動先を表す to があげられる。なお、命令を実行するエージェントは、属性 subject の値に入る。

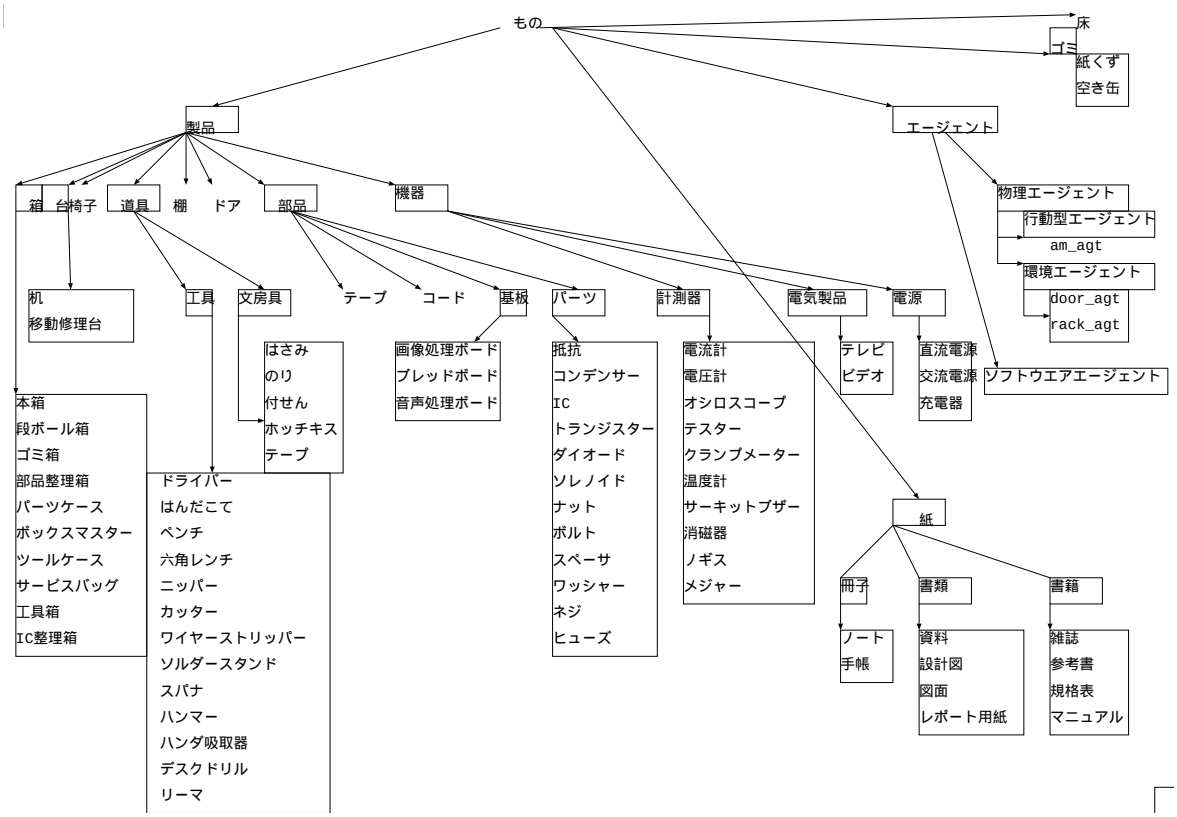

図 7: ワークスペース内の共通知識

一般に、人間が行なう命令は、各エージェントが実行出来るレベルの命令ではなく、基本的な命令が組合わさった複合的な命令である。前者の分解できない命令を基本命令、後者の組合わされた命令を複合命令と定める。なお、基本命令の属性を増やした命令も複合命令と定める。

基本命令の例を以下に記す。

移動する	move
掴む	grasp
放す	release
見付ける	find
維持	hold
開く	open
閉める	close
手渡される	r_handover
手渡す	h_handover

複合命令の例の例を以下に記す。

取って来る	fetch(move+bring)
片付ける	clear(move+take+handover, move+take+release)
持って来る	bring(handover+carry, find+grasp+hold+carry)
持って行く	take(handover+carry, find+grasp+carry)
探す	search(find,move+find)
案内する	guide(search+move)
手渡す	handover (r_handover+h_handover, move+find+grasp+carry +release)
運ぶ	carry(move)
来る	come(move)
行く	go(move)

なお、() 内は基本命令及び複合命令の組み合わせを表す。

5 メディエータの機能

先に述べたように、異なるエージェント間での知識の伝達による協調を実現するための機構がエージェントネットワークには必要であり、この機構を持つエージェントをメディエータと呼ぶ。以下では、特にタスクからの行動の生成という側面からメディエータの機能を明らかにする。

ある抽象的な記述でタスクが与えられた時、最終的には各エージェントに対する命令の系列を生成しなければならない。(図 8参照)。このためにまず、そのタスクにかかれるものを正しく認識する必要がある。これを属性値の付加機能と呼ぶ。当然、タスクはエージェントで実行可能になるまで分解する必要がある (タスク分解機能)。同時に分解された各タスクをどう各エージェントに割り振るかということも遂行しなければならない (エージェント割り当て機能)。また、各々のタスクはそ

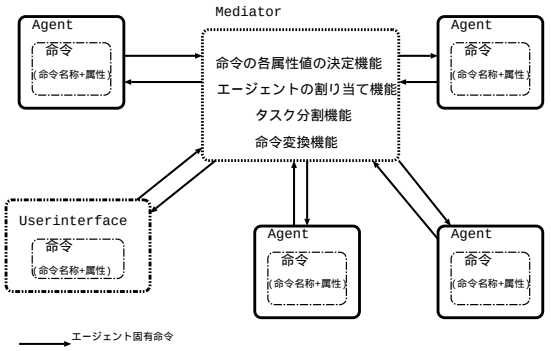

図 8: メディエータの機能概念図

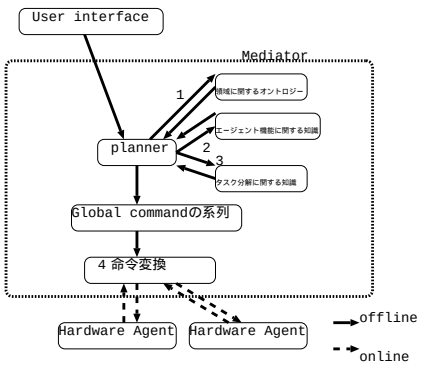

図 9: メディエータの構成図

れが実行されるエージェントに理解可能な記述にする必要がある (命令変換機能)。具体的には、以下の手順で行なわれる (図 9)。

5.1 属性値の付加機能

与えられたタスクを実行する際に、タスク達成に必要な属性が与えられていない場合に、領域に関する共通オントロジーから得る。例えば、扱うものの重さが必要であるが、ものの名称しか与えられていない場合には、そのものが重いものか軽いものかの情報を対象物のクラスの属性を用いて導出する。

5.2 エージェントの割り当て機能

タスクを実行できるエージェントの適切な選択を行なうために、各エージェントの能力の管理を行なう必要がある。各エージェントのハードウェアの仕様および、与えられたタスクに関する知識とを参照して、タスクを実行出来るエージェントを決定する。

5.3 タスク分解機能

与えられるタスクは、命令で表現される。命令が複合命令で、実行できるエージェントがない場合には、命令は基本命令まで分解される。

5.4 共通命令とエージェント固有の命令間の変換

知識にエージェント固有の知識と共有可能な知識があるように、命令にもエージェント間で相互理解できる共通の命令とエージェント固有の命令がある。システムに入力される命令は、共通命令に変換された後にタスク分解で分解され、共通命令系列となる。各命令は、さらにエージェントが理解できるエージェント固有の命令に変換される。命令の変換には、名称の変換と命令の属性の変換が必要である。命令の名称の変換は、共通命令とエージェント固有の命令の変換規則で実現している。命令の属性値の変換は、共有可能な知識とエージェント固有の知識との変換規則で実現している。この変換機能を、各エージェントが分散して持つことも可能であるが、異なるエージェント間では、知識の伝達による相互理解が困難であるという前提なので、メディエータがこの変換機能を有することにした。

6 メディエータの実装

前章で述べたメディエータのプロトタイプを作成した。以下に作成したメディエータの各機能および手順を記す。

1. 命令の属性値の付加機能
   あらかじめ命令の属性が属性間で制約を持つ場合には、制約条件から属性値を決定する。制約条件からも求まらない場合には、領域に関する共通オントロジーを参照してそのクラスあるいはその上位クラスの属性値を用いて命令の属性値を決定する。
2. エージェントの割り当て機能
   各命令の属性値のクラスを比較することによりエージェントを割り当てる。
3. タスク分割機能
   実行できるエージェントが、存在するまで命令を分割する。
   3-1. 命令パターンの選択
   あらかじめ蓄えられた、命令を分割するために必要な知識を有する。fetch 命令を例に命令分解に関する知識について以下に記す。

(分解前)

```
fetch
  :object       object1
  :from_place   on(agent2 or place2)
  :to_place     here(operator1)
  :subject      agent1
  :from         at(agent1)
  :to           here(operator1)
```

(分解後)

```
move
  :subject      agent1
  :from         at(agent1 or place1)
  :to           in front of(agent2
                            or place2)
bring
  :object       object1
  :from_place   on(agent2 or place2)
  :to_place     here(operator1)
  :subject      agent1
  :from         in front of(agent2
                             or place2)
```

```
 :to          in front of(operator1)
```

ここで object1,agent1,agent2,place2, operator1 は変数である。
また、各属性間の制約は同一変数名により表す。

3-2. 命令の属性値の入力
命令の属性値を決める。命令間の引数の制約および各命令内部の引数間の制約により引数の値を決める。

3-3. エージェントの割当
命令を実行できるエージェントを見付け出す。成功すれば終了し、失敗すれば命令を分割するプロセスに戻る。以下に、 move 命令に対するエージェントの割り当てルールを示す。

```
move
 :subject agent
 :from    at(agent)
 :to      in front of(空間指標)

    ----> kappa1b,kappa1a,human
```

なお、空間指標は変数 agent は型である。

例えば、 fetch 命令を実行できるエージェントはないので、 move 命令と bring 命令に分解される。以下にその例を示す。

(分解前)

```
fetch
 :object         solaris
 :from_place     on(rack_agt1)
 :to_place       in front of(operator1)
 :subject        kappa1b
 :from           at(kappa1b)
 :to             in front of(operator1)
```

(分解後)

```
move
 :subject        kappa1b
 :from           at(kappa1b)
 :to             in front of(rack_agt1)
bring
 :object         solaris
 :from_place     in front of(rack_agt1)
 :to_place       in front of(operator1)
 :subject        kappa1b
 :from           in front of(rack_agt1)
 :to             in front of(operator1)
```

なお、 bring はさらに分解される。

4. 共通命令のエージェント固有命令への変換手法
move 命令の属性である移動に関する in front of(rack_agt1) をその命令を実行するエージェントが理解できる地点に関する知識に変換する必要がある。

```
共通命令
move
 :subject        kappa1b
 :from           at(kappa1b)
 :to             in front of(rack_agt1)
```

```
エージェント固有命令
move
 :wait_point     waitpoint 番号
```

7 例題

今、操作者が棚エージェントにある solaris のマニュアルが手元に欲しい状況であるとする。なお、棚エージェントの名称は rack_agt1 である。

- 入力
ユーザインタフェースから solaris のマニュアルを取って来て欲しいことを入力する。その命令を以下に記す。

```
fetch
 :object       solaris
 :from_place   on(rack_agt1)
 :to_place     in front of(operator1)
 :subject        -----
 :from           -----
 :to             -----
```

なお、 ---- は属性値が未入力であることを示す。

- 属性値の付加及びエージェントの割当
fetch を実行できるエージェントはないので、タスク分割を行なう。
- 共通命令系列
得られた共通命令系列を以下に記す。

```
move
 :subject  kappa1b
 :from     at(kappa1b)
 :to       meetingpoint
                 (kappa1b,rack_agt1)
r_handover
 :object     solaris
 :from_place on(rack_agt1)
 :to_place   in front of(operator1)
 :subject    kappa1b
 :from       meetingpoint
                 (kappa1b,rack_agt1)
 :to         meetingpoint
                 (kappa1b,rack_agt1)
h_handover
 :object     solaris
 :from_place on(rack_agt1)
 :to_place   in front of(operator1)
 :subject    rack_agt1
 :from       meetingpoint
                 (kappa1b,rack_agt1)
 :to         meetingpoint
                 (kappa1b,rack_agt1)
carry
 :object        solaris
 :from\_{}place on(kappa1b)
 :to\_{}place   in front of(operator1)
 :subject       kappa1b
 :from          meetingpoint
                   (kappa1b,rack_agt1)
 :to            in front of(operator1)
```

- エージェント固有命令系列
共通命令系列が作成された後に、各命令一つ一つ

がエージェント固有の命令に変換される。得られたエージェント固有の命令の順序を以下に記す。

1. kappa1b に対するエージェント固有命令
   move
   `:wait_point  w3`
   w3 は waitpoint を表す。
2. kappa1b に対するエージェント固有命令
   move
   `:wait_point  w3`
   w3 は waitpoint を表す。
3. 自律型棚に対するエージェント固有命令
   toridasi
   `:object  solaris`
   `:cell_no  3`
   `:point_no  20`
   3 は棚の収納場所の Cell No. を表す。
   20 は搬送テーブルの位置である Point No. を表す。
4. kappa1b に対するエージェント固有命令
   move
   `:wait_point  w1`
   w1 は waitpoint を表す。

8 考察

プロトタイプシステムの実装とその実行結果から考えられるメディエータが持つ各機能についての効果を以下に記す。

- 属性値の付加機能の効果
  今回実験した入力では、取って来て欲しいもの存在場所を事前に知っていることを仮定している。よって、領域に関する共通オントロジーから得る情報の効果は、評価することが出来なかった。今後、領域に関する共通オントロジーによる属性付加による効果について評価して行くつもりである。また、開いた空間である実世界において、すべてのものを管理することは困難である。また、閉じた空間を仮定しても、人と共存する場合でのものの完全な管理は、計算機では予期できない人の行動を管理する必要があり困難である。よって、画像処理などによる実世界からの情報を獲得する必要があり、この実装も予定している。また、エージェントの機能によって、実世界から得る情報の差異は大きいと考えられ、領域に関する共通オントロジーによるものの属性付加機能の効果をこの面からの評価しようと考えている。
- エージェントの割当機能の効果
  共通命令でプランニングする場合には、トップダウンにプランニングするので、エージェントの割当機能は欠かせない。
- タスク分割機能の効果
  単体のエージェントでは困難であったタスクも、柔軟に複数のエージェントが組み合わせることで実行可能になる場合もある。今回の実験では、搬送機能しかない行動型エージェントが棚からものを受け取ることができその有効性を確認した。
- 命令変換機能による効果
  各エージェントがメディエータ機能を持ち、共通命令を直接受け取る方法と、集中型のメディエータから各エージェントが、エージェント固有の命令を受け取る方法がある。今回は、メディエータが命令の変換を行ない、各エージェントは、エージェント固有の命令を受け取る集中型の構造で構築した。しかし、分散型のメディエータの場合は、エージェント側に自律性を持たすことができる。例えば、エージェント間でものを手渡す協調動作のプランニングを行なう場合には、手渡す場所をエージェント間で決定することが出来る。よって、より柔軟な協調動作が可能になると考えられるので分散型のメディエータの構築についても、今後考慮していくつもりである。

以上のことから、メディエータにより異なるエージェントが持つ知識の相互理解を実現するには、領域に関する共通オントロジーによる、ものの属性情報の付加機能、および、命令の変換の様に、エージェント固有の知識とエージェント間で共有可能な知識との変換機能が必要であることがわかった。

9 まとめと今後の展望

異なるエージェント間で、通信により協調動作を行なう場合には、互いに内部に有している知識の相互理解が必要である。本論文では、メディエータを用いることで、異なるエージェント間で知識の伝達が可能であることを、協調支援環境での協調動作のプランニングを行なうことで確認した。今後は、各エージェントが実世界から実際に獲得した知識をエージェント間で共有可能な知識に変換し、エージェント間の協調動作に発展させていきたいと考えている。

参考文献

[1] 松下温、岡田謙一ほか. 図解グループウェア入門. オーム社, 1991.

[2] 堂下、西田、山田. 空間認知と言語理解. オーム社, 1991.

[3] Tim Finin, Don McKay, Rich Fritzson, and Robin McEntire. KQML: An information and knowledge exchange protocol. In Kazuhiro Fuchi and Toshio Yokoi, editors, *Knowledge Building and Knowledge Sharing*. Ohmsha and IOS Press, 1994.

[4] 武田 英明、飯野 健二、西田豊明. 知識コミュニティにおける仲介機能. マルチエージェントと協調計算３. 近代科学社, 1993.

[5] Johann Schweiger. Facilitating teamwork of autonomous systems with a distributed real-time knowledge base. In *IEEE International Conference on Robotics and Automation Volume4*, pp. 2883–2888. IEEE Computer Society Press, 1994.

[6] 長田正偏. 自律分散をめざすロボットシステム. オーム社, 1995.

[7] Jose M. Vidal and Edmund H. Durfee. Task planning agents in the umdl. November 1995.