Vol.2011-MPS-85 No.10
2011/9/15

実世界の多様性に適応したBDIロボットについて

藤田 恵[†1] 片山寛子[†1]
新出尚之[†2] 高田司郎[†3]

近年、単一の目標達成に特化されたロボットでなく、動的に変化する環境のもとで自らの目的を達成するよう動作するロボットの実現が重要な研究課題となってきている。我々は、BDI モデルを実装した BDI ロボットによるロボット制御が、複雑で多様性に富んだ実世界において自らの目的を実現していくロボットのためのアーキテクチャとして有効であることを示す。特に、意図の保持と破棄という機構によって、目標を達成する手段を柔軟に切り替えたり、複数の目的を並行に保持し整合的に実行したりできる BDI モデルの能力が、多様性への適応に非常に有用であることを論じ、また、実験を通じてそれを示す。

BDI Robots who Adapt the Diversity of the Real World

MEGUMI FUJITA,[†1] HIROKO KATAYAMA,[†1] NAOYUKI NIDE[†2]
and SHIRO TAKATA[†3]

Recently, it has been considered important issue to realize robots which acommplish their purposes dynamically in the real world environments. We propose that the BDI robots who implement the BDI model are profitable for such environments because they can acommplish their purposes dynamically. Indeed, the mechanism of maintaining and discarding the robots' intentions can switch their methods for achieving their goals flexibly, and can maintain consistency of concurrent actions. Therefore, we claim that these abilities of the BDI model are profitable for adapting the diversity of the real world, and show them by the experimentation.

†1 奈良女子大学大学院
Graduate School of Humanities and Sciences,Nara Women's University
†2 奈良女子大学
Nara Women's University
†3 近畿大学
Kinki University

1. はじめに

これまでに実用化されているロボットの多くは、単一の目標達成に特化されたもので、たとえば二足歩行エンターテーメントロボットなどは実世界の多様性を考慮に入れることなく、つまり多様な環境変化に適応することなく、如何にダンスなどの振る舞いを上手く表現するかに重点が置かれている。また、振る舞いを実現するために、アクションの状態遷移を網羅してハードコーディングされているものが多い。このようなロボットは、確かに身体を持つが、たとえば空き缶を拾って環境をきれいにするなど、環境と相互作用するために身体を持っている訳ではない。つまり実世界を一定の仮想世界と仮定してロボット制御が行われている。一方、人間と協調的に日常生活を行うことを指向したヒューマノイド型ロボットなどは，人間や環境の多様な変化に適応して、複数の目的を整合的に達成していく必要がある。たとえば看護ロボットなどでは、患者から飲み物を要求されたので冷蔵庫に向かうが、向かっている途中に患者がベッドから落ちてしまった場合、飲み物を後回しにして倒れ込んだ患者をベッドに戻すことを優先する必要がある。さらに、患者に飲み物が欲しいか聞き直して、もし飲みたいという意図を引き続き持っていれば冷蔵庫に向わなければならない。後者のように、矛盾無く達成すべき目的の数が増えれば増える程、それら状態遷移の記述すら困難なことは容易に想像される。

我々は、2 節にて紹介する意図の理論をベースにした BDI モデル[8] に基づいたロボット制御が、多様性に富む実世界において複数の整合的な目的を実現するには有効であると考えている。そこで本論文では、BDI モデルを実装したロボットを BDI ロボットと呼び、実世界の多様性に適応して、複数の目的を整合的に実現するには、BDI ロボットが有効であることを実験を通じて示す。

以下、本論文では、2 節にて実世界における BDI モデルについて、3 節にて今回実験に用いる BDI ロボットの仕様について、4 節にて実世界に多様性への適用実験の内容と実験結果について、それぞれ述べ、最後に、おわりにでまとめる。

2. 実世界の BDI モデル

2.1 実世界のさまざまな問題

身体性を有する実世界のエージェントが長い期間生き延びて機能するためには、仮想世界とは異なる以下のような問題が生じる[6]。

(1) 情報獲得に時間がかかる。たとえば、隣の部屋にプリンタがあるのかどうか知りたけれ

ば、行ってみるか、だれかに聞くか、あるいは、ネットワーク構成図をみなければならない。

(2) 極めて限られた情報しか得られない。たとえば、サッカーゲームにおいては自分の視界の範囲の情報しか得られない。

(3) 物理デバイスは外乱や故障から逃れられない。

(4) 実世界を離散的には記述できない。

(5) 実世界のエージェントは常に複数のことを並行に行う必要がある。たとえば看護ロボットは、患者の話し相手をしながら、飲み物を運んだり、ベットに寝かせたり、体温や脈をはかりながら主治医へ連絡するなど、患者に状況や要求に適応して、複数の目的を並行して行う必要がある。

(6) 実世界は固有のダイナミックスで常に変化している。

(7) 実世界は極めて複雑なダイナミカルシステムであり、その非線形特性と初期値に対する鋭敏性ゆえに、本質的に予測不可能である。

そこで我々は、このような実世界の特性に適応するために、BDI モデルのベースである Bratman の意図の理論[2] に立ち返る。意図の理論とは、次に述べるように実世界における合理的エージェントの意図に関する哲学的な分析である。

2.2 意図の理論

意図の理論によれば，われわれ人間は、非常に複雑で常に変化する実世界において、複数の目的を達成するために計画立案する合理的エージェントである。われわれは、目的を達成するためには、まずどのような様態 (目標) を達成すべきであるかを決定し、次に、その目標をどのような手段を用いて達成するかを決定する。前者を熟考、後者を目的-手段推論と呼び、これら二つを合わせて実践的推論と呼ぶ。また、現在形成されている複数の意図の中から、どの意図を次に実行するか推論することも熟考と呼ぶ。

われわれは、実世界では、計画立案に必要な信念 (belief) を全ては保持できない (信念の不完全性)。これら信念から未来の予測を全て計算することはできない (推論能力の有限性)。また、その都度の状況に応じて瞬時に判断を下して行為を繰り返すことで複雑な目的を達成することは難しい。このように、われわれの能力は限られたものであり、未来において行うこと全てを特定するプランを一挙に立てることは不可能である。また、われわれが住む実世界は常に変化しているため予期せぬ事態が発生し、事前に立てた計画は全く無駄に終わるかもしれない。

そこで、われわれは最初からプラン本体を全て具体的に実行できる行為列で埋めたプランではなく、環境の変化に柔軟に適応できる程度の粒度からなる副目標を用いたプランを立案した後、そのプランを意図として形成した後、実行に移す。そして、まだ埋められていない副目標を実行する時が来たと判断したとき、その状況に適応した、副目標を実現する (サブ) プランを目的-手段推論し、そのプラン本体を手段として実行する。このように状況の予測が容易でないときは、副目標の実現手段の推論を先送りすることで、多様な環境変化に柔軟に適応した意思決定を行うことができる。

2.3 意図の再考慮とコミットメント戦略

また、意図の理論において、意図の再考慮が合理的であるのは、以下の場合である。

(1) 計画時に想定した環境の状況と現状とに相違点がある場合。つまり、計画した時点で誤った信念を持っていたことに気が付いた、または、現状が計画時の予想とは異なる場合

(2) エージェントの欲求または価値意識に問題となるような変化があった場合

(3) 他の意図のいくつかに問題となるような変化があった場合。つまり、意図間の整合性が取れなくなった場合

である。

また、意図の持続と破棄に関する「コミットメント戦略」[8] について紹介する。

(a) Blind(盲目的) エージェントは、意図はすでに実現されていると信じるまで、その意図を持続する。

(b) Single-minded(一意専心) 意図はすでに実現されていると信じるか、もしくは、その意図の実現が可能であると信じなくなるまで、その意図を持続する。

(c) Open-minded (心の広い) その意図を形成した欲求を実現するという状況でなくなるまで、意図を持続する。つまり、その意図はすでに実現されたと信じるか、その欲求を実現する理由がなくなったので取り下げるまで、その意図を持続する。

合理的エージェントはこのようなコミットメント戦略に基づいて意図の持続や破棄を行う。

コミットメント戦略を用いれば、以下のように、基本行為レベルや目標レベルでの再考慮も可能となる。たとえば、基本行為が失敗したときは、その基本行為を再実行するか、またはその基本行為が実行できなければ、現在目指している目標を達成する別の手段を目的-手段推論して、その目標を持続することができる。さらに、この目標の実現手段がなくなれば、目標レベルで再考慮し、意図を実現する別の目標を熟考して、その目標を達成するよう意図を持続する。このような再考慮は意図を持たない (たとえば目標だけを持つ) エージェントで実現することは難しい。

2.4 実世界の緒問題と意図の理論の対応

そこで、実世界の諸問題と意図の理論との対応を考えてみる。まず (1) に対しては、実世界の情報を得るプランを意図として保持し、情報が必要になれば実行することで、実世界と相互作用を行ない所望の情報を得る。または他のエージェントに依頼することで得ることもできる。本論文では、どちらの場合も実験対象とする。

次に (2) に対しては、行為列を実装していない副目標を用いたプランを意図として保持・実行することで、状態が分かるまで保留するという場合が考えられる。また、センサの性能で自力では正確な情報が得られない場合は、正確な情報を得ることができるエージェントに依頼するプランを意図として保持・実行することで得ることができる。本論文では、どちらの場合も実験対象とする。

次に、(5) に対しては、複数の目的を状況に適応して実行するには、それぞれの目的を達成する意図を保持し、並行に実行することができる。本論文では、複数の目的を並行に実行する実験を行う。

3. BDI ロボット

本節以降で、我々は、BDI ロボットが実世界の多様性のもとで複数の目的や意図を整合的に達成できること、特に 2.1 節に述べた実世界における問題のうちの (1), (2), (5) に対処できることを実験で示す。本節では実験に用いたロボットと、それを制御する BDI ロボットの実装について述べる。

我々は今回、2 台のロボット、Explorer と ForkLift (Lift) を用意した。Explorer の目的は、掃除を行うこと、荷物を置く台を発見した場合に Lift に伝えること、Lift に依頼されたときに台の識別を行うことであり、Lift の目的は荷物を置く台に自身のもつ荷物一個をその台に置くことである。

初期状態ではこれらのロボットを制御する BDI ロボットは台がどこにあるかは知らず、台の位置に関する信念はロボットの知覚行為によって加わっていくこととなる。

3.1 ロボットの構成

我々が用意した 2 台のロボットは、LEGO 社が開発した商用教育用のロボット LEGO MINDSTORMS NXT を用いて作成した。NXT は安価で比較的制御が簡単であり、また、たくさんのパーツを組み立てて一つのロボットを作成するためロボットの形状はさまざまなものに仕上げることが可能である[3] ので、実験に用いるには適したロボットである。2 台の作成にあたっては NXT の組み立て用 Web ページ[5] を参考にした。2 台の主な特徴を以下に述べる。

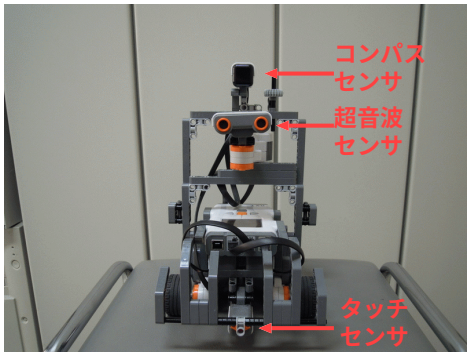

図 1 Explorer 正面図

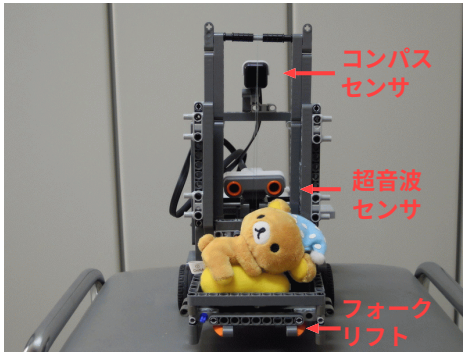

図 2 Lift 正面図

- Explorer
  Explorer は図 1 のような形状をしたロボットであり、
  - コンパスセンサ
  - 超音波センサ
  - タッチセンサ

  を持つ。
- Fork Lift
  Fork Lift (以下、Lift) は図 2 のような形状をしたロボットであり、
  - コンパスセンサ
  - 超音波センサ
  - フォークリフト

  を持つ。

3.2 ロボットの基本行動

ロボットの基本行為としては、グリッドワールドでの行動制御をにらみ、次のようなものを用意する。これには Python によるライブラリ NXT_Python[4] を用いている。

- 1 マス前進
- コンパスセンサを用いた現在の方向の知覚、及び 90 度単位の回転 (右、左)

- 超音波センサによるロボットの前方方向への知覚
- タッチセンサによる衝突認識 (Explorer のみ)
- フォークリフトの上にあらかじめ載っている箱を台の上に置く (Lift のみ)

3.3 環境設定

実験の環境設定は以下のものである。

- 実世界上にグリッドワールドを設置 (図 4 は配置の初期状態の例で、大きさが 5×4、左上のマスの座標が $(0,0)$、右下が $(4,3)$)
- Explorer と Lift と台をグリッドワールド上のマス目に配置
- 台は、荷物を置く台 (stand) と障害物 (obstacle) の 2 種類がある
- 荷物を置く台は 1 個だけ、障害物は複数個

また、以下の理由により Lift は台を見つけた場合、それが荷物を置く台であるかどうかの判定は Explorer に委託する必要がある (図 3)。これらは 2.1 節の (1) や (2) への対応を示すための設定である。

- Explorer の超音波センサは荷物を置く台より高いため、障害物のみ認識できるが、タッチセンサは荷物を置く台を認識できるため、両センサの併用で台の種類を見分けられる
- Lift は超音波センサのみ持ち、それは荷物を置く台より低いため、台を認識はできるが、台の種類を見分けることはできない

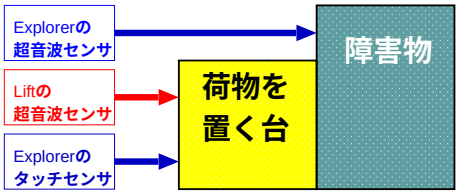

図 3 ロボットのセンサと台の高さの関係

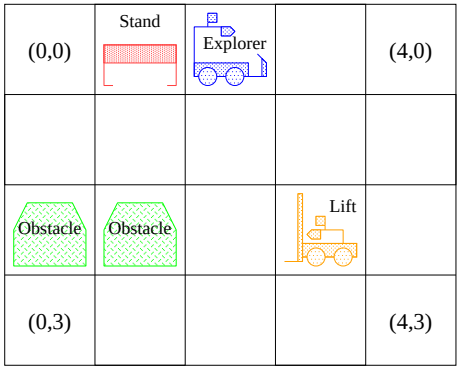

図 4 実験の環境

3.4 Jason

我々は、2 台のロボットが 3.3 節に述べた環境設定におけるそれぞれの目的を達成するよう、これらを制御する BDI ロボットを実装した。実装には、BDI モデルに基づくエージェントアーキテクチャである BDI アーキテクチャ[9] の実装プラットフォーム、Jason[1] を用いた。まず本節では Jason について述べ、次節以降でロボットを制御する BDI エージェントの実装について述べる。

3.4.1 Jason インタプリタ

Jason[1] とは、BDI アーキテクチャに基づくエージェント記述言語 AgentSpeak[7](の拡張) の処理系である。Jason では、エージェントのプランや信念は Prolog に似た文法を持つルールの形で宣言的に記述し、環境モデルの定義は Java 言語で行う。後者にはエージェントと環境の相互作用に関する記述、特にエージェントの環境知覚や基本行為の定義も含まれる。定義された環境知覚や基本行為はプラン内で用いることができる。

プラン記述は基本的に

目標
: プランを適用するための前提条件
<- プラン本体 (基本行為または副目標の列)

という形をとる。プラン頭部の「目標」は実際には、その目標を生じさせるイベントの発生として記述される。Jason インタプリタは、イベントが発生していれば、それに合致する頭部を持つプランの中から 1 つ選んで (「目的-手段推論」)、エージェントの意図の集合に追加する。次いで現在の意図のうち 1 つを選んで (「熟考」)、その本体の行為 1 つ (基本行為または副目標) を実行する。これを繰り返す。

3.5 BDI ロボットの目的と目標

2.3 節で述べた意図の再考慮やコミットメント戦略などの振る舞いを実現するために、BDI ロボットの目的及び目標を記述する。各目標の最後に書かれている「(プラン名)」は、それぞれの目標を実現するための、3.6 節にて記述されているプランの名前である。

まず、Explorer の目的は以下のように 3 つあり、それぞれの目標とともに列挙する。「●」で始まるものは目的、「–」で始まるものはその目的を達成するための目標である。

- 掃除をくまなく行うこと
  - 電池が消耗するまでの間、グリッド内を掃除をして回る[⋆1](プラン (a))

⋆1 実験では実際は単に動き回っているだけだが、モップか何かを持って清掃して回っているという想定である。

  - 電池が消耗してきた時、掃除のタスクは終了され初期位置に戻る (プラン (d))
- 荷物を置く台を発見したら Lift に伝えること
  - 掃除をしながらくまなく移動している途中に荷物を置く台をたまたま発見した場合は、Lift に台を発見したことを伝える (プラン (c))
- 荷物を置く台かどうかの識別を行うこと
  - Lift に台の識別を依頼された場合は最優先して台の識別を行う (プラン (b))

次に、Lift の目的は 1 つである。「*」で始まるものは副目標である。

- 荷物を置く台に一個荷物を置くこと
  - 荷物を台に置く目的を達成するために自身の持つ荷物を荷物を置く台に載せる (プラン (e))
    * まず荷物を置く台の探索を行う
    * 台の発見と共に Explorer に台の識別を依頼
    * 台に荷物をおくことができたならば、タスク終了とみなし、初期位置に戻る

    この目標を達成するために 2.3 節で述べた失敗した場合の処理を考え、意図の保持によって柔軟に対処することが可能であることを示すために、次のような失敗にも対処可能としている。
    * 環境の変化による失敗
      (1) Explorer が先に荷物を置く台を発見した場合に、Lift にそれを知らせる (プラン (c)、プラン (g))
      (2) Lift がその台の位置に向かう間に台がなくなってしまうことにより (環境の変化)、荷物を置くという目標の達成に失敗する (プラン (g) における失敗)
  - Explorer から荷物を置く台の発見を知らされた場合には、その場所に向かい荷物を置く (プラン (g))
  - Explorer の電池が消耗してきた時、Explorer はすべてのタスクを放棄するため、自身の依頼も受け入れてもらえなくなることから、Lift は荷物を置くタスクを達成不能と判断し、荷物を置くことをあきらめる (プラン (i))
  - Lift が現状の探索の範囲が偏りがあると判断したとき、探索の戦略を変更する。(プラン (h))

3.6 BDI ロボットの目標を達成するプラン

上に述べた BDI ロボットの目標を達成するプランを以下に擬似コードで記す[*1]。

- Explorer のプラン
  (a) くまなく掃除をするプラン

```
clean_around
:  true
<- look_around; // 周囲のマスの知覚
   move; // 動けるマスを1つ選んでそこへ移動
   clean_around. // 再帰
```

  (b) 台の識別をするプラン ((a) より優先)

```
identify_rack
:  receive(lift, identify_request(X,Y))
   // Liftから識別依頼が来たら
<- wait(current_pos(X1,Y1)); // 自分の現在位置の取得
   calculate_route_to_next(X1,Y1,X,Y,Route);
   // (X,Y)の隣までの経路の計算
   move_along(Route); // 移動
   judge(X,Y,Type); // 識別
   send(lift, type(X,Y,Type)). // Liftに結果を伝える
```

  (c) 荷物を置く台を発見したことを Lift に伝えるプラン

```
notify_of_stand
:  perceive_touch(X,Y) // タッチセンサが感知
<- if(percept(X,Y)){ // 超音波センサで知覚し直す
     // 知覚があればそれは障害物
   } else { // 何も知覚できなければ荷物を置く台と判断
     send(lift, type(X,Y,stand)). // Liftに結果を伝える
   }
```

  (d) 電池が消耗してきて意図を放棄するプラン

```
abandon
```

*1 Jason で用いられる、イベントの追加などを表す記号 (「+!」など) は省いた。

```
   :  timeout // 電池消耗により時間切れ
   <- drop_all_desires; // すべての願望を放棄
      send(lift, abort); // Liftに放棄したことを告げる
      return. // 初期位置に戻る
```

- Lift のプラン

(e) 荷物を台に置くプラン

```
put_baggage
:  true
<- search_stand(X,Y); // 荷物を置く台を見つける
   if(percept(X,Y)){ // 念のため超音波センサで知覚し直す
     put_baggage_at(X,Y); // 荷物を置く
     return; // 初期位置に戻る
   } else {
     fail; // 失敗
   }
```

(f) 荷物を置く台を見つけるプラン (自分で見つける場合)

```
search_stand(X,Y)
:  not already_put_baggage // 荷物をまだ置いていない
<- find_object(X1,Y1); // 台を探す
   send(explorer, identify_request(X1,Y1));
   // explorerに識別を依頼
   wait(type(X1,Y1,Type));
   if(Type != stand){ // 荷物を置く台ではなかった
     search_stand(X,Y); // 再帰
   } else {
     X=X1; Y=Y1; // 荷物を置く台があった
   }
```

(g) 荷物を置く台を見つけるプラン (Explorer に教えてもらう場合。(f) より優先)

```
search_stand(X,Y)
:  receive(explorer, type(X,Y,stand))
<- wait(current_pos(X1,Y1)); // 自分の現在位置の取得
   calculate_route_to_next(X1,Y1,X,Y,Route);
   // (X,Y)の隣までの経路の計算
   move_along(Route). // 移動
```

(h) マップの左半分優先で台を探すプラン ((f) のサブプラン。右半分優先の同様のプランもある)

```
find_object(X,Y)
:  more_unseen_places_on_lefthalf // 未見の地が左半分に多い
<- set_search_strategy(left_prior);
   // 探索方針変更; look_aroundやmoveに影響を及ぼす
   look_around; // 周囲のマスの知覚
   if(found_object(X,Y)){ // 台があった
     // そのまま終了
   } else {
     move; // 動けるマスを1つ選んでそこへ移動
     find_object(X,Y); // 再帰
   }
}
```

(i) Explorer の電池が消耗してきた時に Lift がとるプラン

```
give_up
:  receive(explorer, abort)
<- wait(current_pos(X,Y)); // 自分の現在位置の取得
   drop_all_desires; // すべての願望を放棄
   return; // 初期位置に戻る
```

これらの中で用いられている `move` や `look_forward` などのサブプランは、最終的には 3.2 で述べた基本行為 (と Jason の内部アクション) を用いて実現されている[⋆1]。

4. 実世界の多様性への適応実験

3.3 節で述べたような設定のもと、実際に我々のロボットが実世界の多様性に適応できる

⋆1 例えば move は、基本行為の「1 マス前進」と、Jason の内部アクションである他エージェントとの信念のやりとりなどを用いて実現されており、自分が進もうとしているマスの占有を他のエージェントに伝えることで、エージェント同士の衝突を防ぐように作られている。

ことを示すため、以下に述べるような 5 種類の配置および状況のもとで実験を行った。

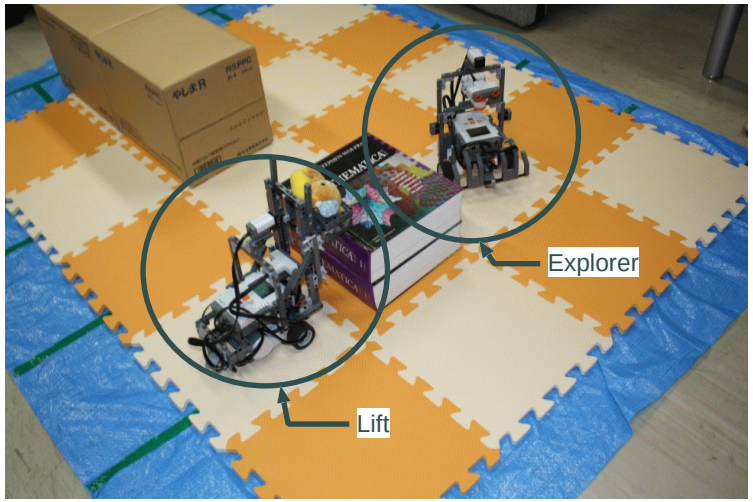

図 5 実験の様子

- 図 6 の配置
- 図 7 の配置
- 図 8 の配置
- 図 6 の配置だが、Explorer が Lift に台の発見を伝えた後に台が移動してしまうという状況 (3.5 節で述べた環境の変化が起きるケース)
- 図 9 の配置

各実験の目的および結果を表 1 でまとめる。図 5 は図 7 の配置からスタートした場合の実験の 1 場面 (Lift が台に荷物を置いているところ) である。

5. おわりに

本論文では、BDI モデルによる実世界ロボットの実現の有効性について、実験を通じて示した。BDI モデルは、意図の保持と破棄という機構によって、複雑に変化する多様性に富む実世界において、目標を達成する手段を柔軟に切り替えたり複数の目的を並行に保持したりできる。また、意図の管理が、タスクスケジュールに相当し、優先順位や整合性を維持しつ

表 1 実験の結果の表

実験の目的	初期配置	観測されたロボットの行動 (括弧内は 3.6 節の該当プランの番号)	評価
意図の再考慮 (2.3 節) が行えること	図 6	Explorer は掃除を開始 (a) したが、(1,0) に荷物を置く台を発見し、Lift に伝えて (c) 掃除を再開。Lift は荷物を台に置くプラン (e) を開始していたが、台を見つける副目標のための意図 (f) を中断し、Explorer から伝えられた台のところへ行く (g) ことで同目標を達成。(e) の残りを実行して荷物を置き、初期位置に戻った	Lift が予期と異なる状況に直面し、目標を達成するための意図を切り替えることができたことで、意図の再考慮が行えることを示した
情報の不完全さ (2.3 節の (2)) や時間がかかること (同 (1)) への対応、および複数の意図の並行 (同 (5))	図 7	Lift は (3,2) の台を発見 (f) し、Explorer に識別を依頼。Explorer は掃除のタスク (a) を中断し、プラン (b) によって台の識別を行った。その結果、荷物を置く台であることがわかり、それを Lift に伝えて (b) は完了、(a) を再開した。Lift は荷物を置く台だったことを知って (f) を完了、台を見つける副目標を達成。以後は上と同じ	Lift は、台を見つけても識別はできない情報の不完全さ、および識別情報が来るまで時間がかかる事象に対処した。また、Explorer は掃除の意図を保持したまま台の識別の意図も達成しており、こちらは複数意図の並行の実現を示している
コミットメント戦略 (2.3 節の (b)、(c)) の実現	図 8	Explorer は (a) の最中、台がないため台を見つられないうちに制限時間がきた。したがって (d) により全タスクを放棄して Lift に通告し、初期位置に戻って終了。Lift は通告を受け、(i) により全タスクを放棄して初期位置に戻り終了	Explorer は探索の実現を放棄して終了したため Open-minded、Lift は荷物を置く目標の達成不能 (台の識別ができないため) により終了したため Single-minded の各コミットメント戦略を実現している
意図の継続 (2.3 節) による失敗への対処が行えること	図 6 の配置で、Explorer が Lift に台の発見を伝えた後に台が移動	一番上と同様に、Lift は Explorer から伝えられた台のところへ行ったが、台がなくなっていて (e) が失敗。しかし、荷物を台に置く目標は残っているので、この目標を達成するために再度プラン (e) が選ばれて意図され、台を探し直し目的を達成した	目標達成の手段が失敗しても、目標が残っていれば新たな手段を選ぶことで意図を持続できることを示した。また、新たに (f) を実行する際、`find_object` の達成手段はその時の状況によって変わる (左/右のどちらを先に探すかはその時になってから選ぶ) ので、この部分は 2.2 節で述べた副目標の実現手段の推論の先送りの例でもある
現状での問題点	図 9	Lift が (3,2) の台の識別を依頼し、Explorer は (b) が (a) を抑止した格好で識別に向かうが、その途中で (3,1) の台を見つけて (c) が割り込む。(b) と (c) の間には排他関係が書かれていなかったため両者が競合し、正常に動作できなくなった	割り込みの入れ子のような場合における割り込み同士の競合には (通常の開発同様) 慎重な検討が必要である

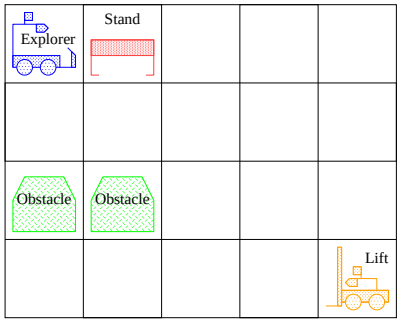

図 6 Explorer による発見の場合の配置図

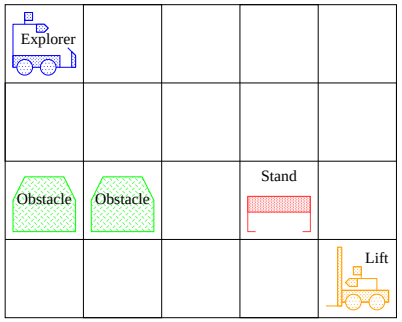

図 7 Lift による発見の場合の配置図

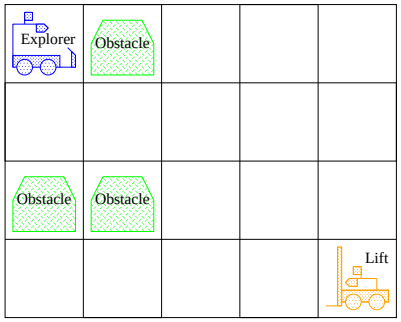

図 8 すべて障害物であるの場合の配置図

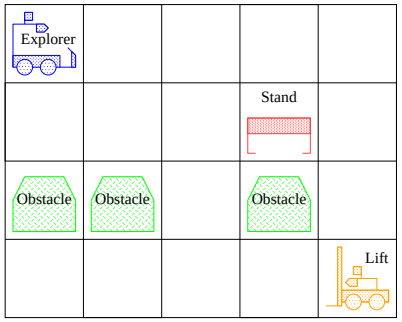

図 9 例外処理が二重に起こる場合の配置図

つタスク管理を行えること、タスク管理をタスク (プラン) 自体と自然に分離できることも、大きな利点である。

今後の課題としては、ロボットの基本行為の精度に関する問題を解消するため、ロボティクス分野の成果によるロボット制御技術との結合による実験の展開や、また、プランニングなどを必要とするより大きな問題へ本論文の手法を適用し、より実用的な課題への応用を図ることなどが考えられる。

さらに、本論文では入手の用意さのため小型ロボットを用いたが、より大規模なロボットへの応用も将来的視野としては考えられる。ただしその場合、実世界で人間など他者と混在した場合の安全性の議論も必要となろう。

参 考 文 献

1) Bordini, R. H., Hübner, J. F. and Wooldridge, M.: *Programming Multi-Agent Systems in AgentSpeak using Jason*, John Wiley & Sons (2007).
2) Bratman, M.E.: *Intention, Plans, and Practical Reason*, Harvard University Press (1987).
3) Claudia and Thomas: LEGO Mindstorms NXT: Welcome to the NeXT generation, `http://www.tik.ee.ethz.ch/tik/education/lectures/PPS/mindstorms/sa_nxt/index.php?page=print` (2006).
4) Lau, D. P.: NXT_Python, `http://home.comcast.net/~dplau/nxt_python/index.html` (2007).
5) Parker, D.: nxtprograms.com, `http://www.nxtprograms.com/index.html`.
6) Pfeifer, R. and Bongard, J. C.: *How the Body Shapes the Way We Think*, The MIT Press (2006).
7) Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language, *Proc. of MAAMAW-96*, LNAI, Vol.1038, Springer-Verlag, pp.42–55 (1996).
8) Rao, A.S. and Georgeff, M.P.: Modeling Rational Agents within a BDI-Architecture, *Proc, of International Conference on Principles of Knowlegde Representation and Reasoning*, pp. 473–484 (1991).
9) Singh, M. P., Rao, A. S. and Georgeff, M. P.: Formal method in DAI: Logic-Based Representation and Reasoning, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, pp.331–376 (1999).