エージェント群の協調作業による機械の組立分解プランニング*

今 村 聡^{*1}

Machine Assembly/Disassembly Planning by Cooperative Agents

Satoshi IMAMURA

A machine assembly/disassembly planning system based on agent cooperation has been developed. It can create an optimal assembly and/or disassembly sequence for any goal, such as the disassembly/assembly of all parts or the replacement of a broken part. The planning system is reactive, i. e., it can cope with unpredictable incidents, such as modification of the goals while the system is operating. We define twenty six class agents for machine assembly/disassembly planning tasks and use an improved algorithm the original of which was proposed by P. Maes. The framework of the proposed planning system and detailed definition of the planning algorithm are introduced and the results of experiments are discussed.

Key Words: Assembly, Disassembly, Process Planning, Production System, Multi Agent Architecture, Topological Relations of Assembly Parts

1. 緒 言

機械のアセンブリプランニングは、設計段階における組立可能性の検証、製造およびメンテナンスにおける組立分解作業のための手順作成など重要な役割をもつ。本研究は、組立分解の作業単位を受けもつエージェント群の協調作業によるアセンブリプランニングに関するものである。

アセンブリプランニングの研究はすでに多くのものが存在する。北嶋⁽¹⁾は有向グラフにより、伊藤⁽²⁾、内山⁽³⁾、De Mello⁽⁴⁾らはアセンブリのプラン木により、関口⁽⁵⁾は部品接続関係の情報を用いて、Seow⁽⁶⁾は時相論理に基づき機械の分解手順のプランニングを行った。いずれのシステムも、最初に与えられた組立モデル空間でプランニングを行うもので、リアルタイムに変化が起こる実世界に適応することができず、柔軟性に欠ける点が課題であった。

これらのシステムに対し、本システムはエージェント群の協調作業を利用し、全体または一部の組立/分解、部品交換などのきめ細かな目標の設定や、突然の目標または組立状態の変更など柔軟性のあるプランニ

ングが可能な点が特徴となる。

協調計算のアルゴリズムは、Maes⁽⁷⁾の提案した方式を改良したものを用いている。これは(1)高い柔軟性：予測されない事態への高度な対応能力、(2)頑強性：エージェント定義の不完全性に対する、ゆるやかな性能劣化(graceful degradation)、(3)モジュール性：新しいエージェントの導入が容易、などの特長をもっているが、一方、(1)ループ：同じ動作列をいつまでも繰り返す、(2)デッドロック：局所最適に陥り、どのエージェントも動作を起こさなくなる、(3)不安定性：状態、目標に応じてシステムパラメータの再調整が必要となる、などの欠点があった。本研究では、Maesのアルゴリズムを機械の組立分解プランニングに適用するにあたり、これらの欠点をなくすために、(1)クラスエージェントの導入、(2)組立ユニットの導入、(3)バイアス活性度の導入、を行った。

以下、協調プランニングの概要、協調アルゴリズム、主軸台を例題とした組立分解プランニングの実験結果について述べる。

2. 協調プランニングの概要

機械の組立分解プランニングシステムは協調プランニング・シミュレータ、大域的データベースおよび26

* 原稿受付 1996年9月11日。

*¹ 正員、機械技術研究所 (〒305 つくば市並木1-2)。

種類のクラスエージェント群から構成される。協調プランニング・シミュレータはエージェント群の動作環境を提供するためのもので、活性度交換計算等を行う。大域的データベースは目標(goal)と状態(state)からなる。状態とは対象機械の組立状態を、face/5(述語名face, 引数の数5の意味。fit/5等も同様)、fit/5 述語(後述)により表現したものである。エージェント群は目標を満たすまで、協調して状態を書き換えていく。クラスエージェントは、図1のフォーマットにより定義される。図2に、実際に定義されたクラスエージェントのいくつかを示す。また、付録にクラスエージェントの定義に用いた述語の意味を示す。

エージェントにはクラスエージェントとインスタンスエージェントがあり、実際に協調プランニングを行うのは問題に応じて生成されるインスタンスエージェント群である。クラスエージェントは生成条件(base-condition), 前提条件(precondition), 否定リスト(negate), 基底活性度(bias)により定義される。生成条件は、クラスエージェントが対象問題に応じて適切なインスタンスエージェント群を生成するための条件である(以降、インスタンスエージェントは単にエージェントと略す)。前提条件は、エージェントが活動するための必要条件である。否定リストはエージェントの活動の内容を規定するもので、書き換えの対象となる述語を指定してある。エージェントはこれらの定義以外に活性度という値をもち、基底活性度はエージェントの活性度の最低値を指定する。エージェントは生成条件を満たせば生成され、生成条件以外のすべての項目をクラスエージェントから継承する。エージェントは前提条件を満たし、最も活性度の高いものが状態を書き換える。

エージェントへの活性度の入力とは状態と目標から与えられる外部入力とエージェント間の活性度交換による。状態からの外部入力は、前提条件と状態に書かれた述語のいくつかが一致するエージェントに対して入力され、その値は前提条件の満たされる割合に比例する。ゴールからの外部入力は、ゴールの述語を生成することができるエージェントに対し、まだそのゴール

述語が実現されていない(大域的データベースにその述語が存在しない)場合に入力される。

エージェント間には活性度伝達を行うための活性度伝達リンクが張られる。これは、先行エージェント(先行して動作するエージェント)が動作することにより後継エージェントの前提条件のfactが満足される状態が生成される場合に張られる。エージェント間の活性度伝達は、後継エージェントから先行エージェントへ流れる。外部入力における状態とゴールからの入力の割合、活性度伝達の比率はシステムパラメータとして調整できる。

大域的データベースは次の2種類の述語により表現される。

fit(Dir, Part1, Part2, Fit Type, Truth)

```

agent screw_a_bolt_to_right(Dir,Name)
base_condition
[ bolt(Name), face(Dir,_,Name,cl,_) ];
precondition [ fit(Dir,Name1,Name,sc,f),
positioned(Name1), not face(Dir,Name,_,_,t),
face(Dir,Name2,Name,_,f), positioned(Name2) ];
negate
[ all fit(Dir,X,Name,_,f),
all face(Dir,_,Name,_,f) ];
bias 1.0; end.

agent loosen_a_nut_to_left(Dir,Name)
base_condition { nut(Name),
fit(Dir,Name,N1,sc,_) , face(Dir,N1,Name) };
precondition
[ fit(Dir,Name,_,sc,t), face(Dir,_,Name,_,t),
dismounted_unit_member(Name),
not ( fit(Dir1,Name,_,sc,t),
Dir1 == Dir), not face(Dir,Name,_,_,t) ];
negate [ all fit(Dir,Name,_,sc,t),
all ( fit(Dir,Key,_,_,t),
call(fit(Dir,Name,Key,ky,t))),
all fit(Dir,Name,Key,_,t),
all face(Dir,_,Name,_,t) ];
bias 0.0; end.

agent push_in_inner_fit_to_right(Dir,Name)
base_condition
[ fit(Dir,Name,N1), face(Dir,N1,Name),
not bolt(N1), not ( face(Dir,Name,N2,_,_),
bolt(N2)), not fit(Dir,_,Name,sc,_) ,
not base(Name), not fit(Dir,Name,_,sc,_) ];
precondition
[ fit(Dir,Name,N2,_,f), positioned(N2),
assembled(Name), not face(Dir,Name,_,_,t),
not ( face(Dir,N,Name,_,f),not positioned(N) ) ];
negate
[ all ( fit(Dir,Name,X,_,f),positioned(X)),
all ( fit(Dir,Y,Name,_,f),positioned(Y)),
all ( face(Dir,Z,Name,_,f),positioned(Z) ) ];
bias 0.1; end.

```

```

agent <Agent Name>
base_condition <Condition for Instanciation>;
precondition <Precondition for activation>;
negate <Predicates to be negated>;
bias <Real number>;
end.

```

Fig.1 Class agent description format

Fig.2 Examples of class agent

Table 1 Class agents for assembly and disassembly tasks

(1) screw_a_bolt_to_right(Dir,Name)	(2) loosen_a_bolt_to_left(Dir,Name)
(3) screw_a_bolt_to_left(Dir,Name)	(4) loosen_a_bolt_to_right(Dir,Name)
(5) screw_a_nut_to_right(Dir,Name)	(6) loosen_a_nut_to_left(Dir,Name)
(7) screw_a_nut_to_left(Dir,Name)	(8) loosen_a_nut_to_right(Dir,Name)
(9) put_unbolted_part_to_left(Dir,Name)	(10) pull_out_unbolted_part_to_right(Dir,Name)
(11) put_unbolted_part_to_right(Dir,Name)	(12) pull_out_unbolted_part_to_left(Dir,Name)
(13) push_in_inner_fit_to_right(Dir,Name)	(14) pull_out_inner_fit_to_left(Dir,Name)
(15) push_in_inner_fit_to_left(Dir,Name)	(16) pull_out_inner_fit_to_right(Dir,Name)
(17) push_in_outer_fit_to_right(Dir,Name)	(18) pull_out_outer_fit_to_left(Dir,Name)
(19) push_in_outer_fit_to_left(Dir,Name)	(20) pull_out_outer_fit_to_right(Dir,Name)
(21) push_in_outer_fit_to_either1(Dir,Name)	(22) pull_out_outer_fit_to_either(Dir,Name)
(23) push_in_outer_fit_to_either2(Dir,Name)	
(24) put_a_key_to_shaft1(Dir,Name)	(25) take_a_key_from_shaft(Dir,Name)
(26) put_a_key_to_shaft2(Dir,Name)	

face(Dir, Part1, Part2, Face Type, Truth)

Dir={*x, y, z*}, Truth={true, false}

Fit Type=

{press, position, movable, taper, screw}

Face Type={contact, gap, taper}

fit/5 は、部品 Part 2 が部品 Part 1 の穴とはめあい関係にあることを表し、face/5 は Part 1 の左側と Part 2 の右側が面していることを表す。Dir は接続関係の方向、Fit Type は、Face Type はそれぞれはめあい、面接関係のタイプを表す。タイプには

pr : しばりばめ

po : 位置決めはめあい

mo : 可動はめあい

ta : テーパー

sc : ねじはめあい

co : 面接触

ga : 面の対向

ta : テーパー接触

がある。Truth は *t* の場合は組立状態、*f* の場合は分解状態を表す。エージェントが活動すると、その negate 内の記述と一致(unify)する fit/5 および face/5 の Turth が *f* から *t*、または *t* から *f* に書き換えられる。

表 1 に示すように、26 種類のクラスエージェントが定義されている。左欄は組立作業、右欄は分解作業用エージェントである。(1)~(8)はねじおよびナットの組立分解用、(9)~(12)はねじ止めされた部品の組立分解用である。(13)~(16)は組立または分解の作業時点で内周はめあい関係のみもつ部品の組立分解用、(17)~(23)は同様に外周はめあいのみもつ部品の組立分解用、(24)~(26)はキーの組立分解用クラスエージェントである。線で結ばれたエージェントのペアは、互いに逆向きの操作を行う関係にあり、base-condi-

tion が等しい。(1)-(3), (5)-(7), (9)-(11), (13)-(15), (17)-(19), (21)-(23), (24)-(26), (2)-(4), (6)-(8), (10)-(12), (14)-(16), (18)-(20)の12種類のペアは、互いに左右対称の関係になっている。すなわち、ペアの一方で現れる face(Dir, P1, P2) および face(Dir, P1, P2, Type, Truth) 述語は、他方では face(Dir, P2, P1) および face(Dir, P2, P1, Type, Truth) の形で現れる (face/3, fit/3 の意味は付録参照)。それ以外の述語はすべて等しい。

組立構造の表現には、ユニット(サブアセンブリ)の概念を導入することができる。ユニットの導入は組立分解手順に一定の制約を与える役割をもつ。ユニットは、unit(Unit, List) 述語で表現され、これは Unit が List に記述された部品群から構成されることを意味する。unit/2 述語が定義されると、大域的データベースは次のルールに従って自動的に書き換えられる。

fit(Dir, P1, P2, Type, TF)

→ fit(Dir, Unit, P2, Type, TF)

fit(Dir, P2, P1, Type, TF)

→ fit(Dir, P2, Unit, Type, TF)

face(Dir, P1, P2, Type, TF)

→ face(Dir, Unit, P2, Type, TF)

face(Dir, P2, P1, Type, TF)

→ face(Dir, P2, Unit, Type, TF)

ただし P1 は Unit 内の部品、P2 は Unit 外の部品である。このような書き換えを行うことにより、分解作業において Unit の取外しは Unit 内のどの部品の取外しよりも先行し、組立作業において Unit 内のすべての部品の組付けは Unit の組付けに先行するようになる。

エージェントの活性度は、大域的データベースからの外部入力と、エージェント間の活性度交換により決まる。外部入力、ゴール(の一部)を実現することの

できるエージェント、並びに、前提条件と状態が部分的に一致するエージェントに対して与えられる。エージェント間の活性度交換は後継エージェントから先行エージェントに対し、その活性度の一部を活性度伝達リンクを通して伝達することにより行われる。なお、活性度伝達リンクとは先行エージェント(後継エージェントの前提条件を満たす働きをもつエージェント)と後継エージェントの間に張られる、活性度交換のためのリンクである。

活性度交換リンクの存在条件は、link(Term1, Term2)述語により与える。ここで、Term1が先行エージェントの否定リスト内の述語、Term2が後継エージェントの前提条件内の述語と一致した場合、二つのエージェント間でリンクが張られる。

エージェントはそれぞれ固有の基底活性度をもつ。これはエージェント間伝達の対象とならない活性度で一定の値をもつ。値はbias部で指定された定数(零以上1.0以下)に、エージェント全体の活性度 π を乗算して決められる。基底活性度が大きいものは、活性度の外部入力や活性度交換の有無にかかわらず、常に大きな活性度をもつので、前提条件さえ満たされれば起動される可能性が高い。例えば、ボルトを締める(1)、(3)およびボルトで固定される部品を取外す(10)、(12)のエージェント(いずれも表1の番号)は基底活性度として1.0の値をもち、前提条件さえ満たせば確実に起動されるようになっている。これはボルトで固定されるべき部品がボルトがはずれた状態で、アセンブリ上に長い時間放置されることを防ぐためのものである。

3. 協調計算アルゴリズム

本章では協調プランニングのための協調計算アルゴリズムを厳密に定義する。次の定義を与える。

エージェント集合： $i=1 \dots n$

Fig. 3 Schematic of a spindle head

$S(j, t)$: 時間 t においてファクト j が真なら 1, 偽ならば零を返す。

$G(j, t)$: 時間 t においてファクト j がゴールのメンバーであり、なおかつ偽ならば 1, それ以外は零を返す。

$B(s, i, k)$: (1) fact k が link(j, k) を満たし, (2) fact k がエージェント s の前提条件の一つを満たし, (3) fact k が真であり, (4) fact j がエージェント i の否定リストの一つと一致すれば 1 を返す。それ以外は零を返す。

bias(i): エージェント i の基底活性度。エージェントの bias 部の値に π を乗算した値

$$0 \leq \text{bias}(i) \leq \pi$$

π : すべてのエージェントの基底活性度を除いた活性度の合計値

β : 後向き活性度伝達の比率 $0 \leq \beta \leq 1$

γ : 外部入力におけるゴールからの入力比率

$$0 \leq \gamma \leq 1$$

時間 t におけるエージェント i への外部入力は,

external(i, t)

$$= \frac{1}{P_i} \sum_{j=1}^{P_i} a(t) \cdot S(j, t) + \frac{1}{g_i} \sum_{k=1}^{g_i} b(t) \cdot G(k, t)$$

.....(1)

ここで、 j, k はそれぞれエージェント i の前提条件内のファクト $1 \dots P_i$, 否定リスト内のファクト $1 \dots g_i$, を表す。 $a(t), b(t)$ は次の式を満たすように決められる。

$$a(t) \sum_j S(j, t) = \pi(1 - \gamma) \dots\dots\dots(2)$$

Table 2 Instance agents for the spindle head

In this table, an instance agent is described by a tuple (Num,Dir,Name), where 'Num' represents a class number given in Table 1. For example, (1,x,29) represents screw_a_bolt_to_right(x,29).

(1, x, 29)	(1, x, 28)	(2, x, 29)	(2, x, 28)
(3, y, 21)	(3, y, 19)	(3, y, 16)	(3, x, 26)
(3, x, 27)	(3, y, 31)	(4, y, 19)	(4, y, 16)
(4, x, 26)	(4, y, 21)	(4, x, 27)	(4, y, 31)
(5, x, 5)	(5, x, 14)	(5, x, 13)	(6, x, 5)
(6, x, 14)	(6, x, 13)	(9, x, 23)	(9, x, 25)
(9, y, 30)	(10, x, 23)	(10, x, 25)	(10, y, 30)
(11, x, 24)	(12, x, 24)	(13, x, 3)	(13, x, 4)
(13, x, 17)	(13, x, 11)	(13, x, 12)	(13, x, 20)
(13, x, 22)	(14, x, 3)	(14, x, 4)	(14, x, 11)
(14, x, 12)	(14, x, 22)	(15, x, 23)	(17, x, 7)
(17, x, 8)	(17, x, 9)	(17, x, 10)	(18, x, 24)
(18, x, 11)	(19, x, 6)	(19, x, 7)	(19, x, 8)
(19, x, 9)	(19, x, 10)	(20, x, 6)	(20, x, 23)
(21, x, 17)	(21, x, 20)	(22, x, 7)	(22, x, 8)
(22, x, 10)	(22, x, 9)	(24, x, 17)	(24, x, 20)
(19, x, unit1)	(20, x, unit1)

Fig. 4 Assembly/disassembly planning simulator

$$b(t) \sum_k G(k, t) = \pi \cdot \gamma \quad \dots\dots\dots (3)$$

ここで i はエージェント 1... n を表す。

エージェント s からエージェント i (s は i の後継エージェント) への後向き活性度伝達は次式で定義される。

$$b(s, i, t) = c(t) \sum_{s_i} B(s, i, s_i) \quad \dots\dots\dots (4)$$

ここで s_i はエージェント s の前提条件のファクトを表す。 $c(t)$ は次式を満たすように決められる。

$$c(t) \sum_p \sum_{s_p} B = a(s, t) \cdot \beta \quad \dots\dots\dots (5)$$

ここで p はエージェント s の先行エージェントを表し、 $a(i, t)$ は時間 t におけるエージェント i の活性度を表す。

次式は時間 t において、エージェント i が後継エージェントから受取る活性度の総和を表す。

$$\text{receives}(i, t) = \sum_s b(s, i, t) \quad \dots\dots\dots (6)$$

時間 t におけるエージェント i の活性度の定義式は次のとおりである。

$$\begin{aligned} a(i, 0) &= 0, \\ a(i, 1) &= \text{external}(i, 1) \cdot (1 - \beta) \\ &\quad + \text{receives}(i, 1) + \text{bias}(i), \\ a(i, t) &= ((\text{external}(i, t) + a(i, t-1) \\ &\quad - \text{bias}(i)) \cdot (1 - \beta) + \text{receives}(i, t)) / 2 \\ &\quad + \text{bias}(i). \quad \dots\dots\dots (7) \end{aligned}$$

エージェントの動作は各時間ごとに次の手順で進められる。

(a) 式(1)によりエージェントへの外部入力を求める。

(b) 式(4)によりエージェント間の活性度伝達を求める。

(c) 式(7)により活性度合計を一定値に調整する。

(d) 次の二つの条件を満たすエージェントが実際に動作を行う。(1)動作可能である、(2)動作可能なエージェントの中で唯一最も高い活性度をもつ。もし、この二条件を満たすエージェントが存在しないときは、何も行わずステップ(a)に戻る。これらのステップ(a)から(d)はすべてのゴールが満たされるまで繰返される。

本アルゴリズムと Maes のアルゴリズムの差は、 $\text{link}(j, k)$ により伝達リンクの接続の仕方を明示したこと、式(7)に $\text{bias}(i)$ を導入したこと、および、前向き活性度伝達リンクを廃止したことである。

4. 機械の組立分解プランニング

数種類の例題に対して実験を行った。実験例として 31 点の部品からなる主軸台を図 3 に示す。この機械の組立構造は 48 個の fit/5 述語と 42 個の face/5 述語により表現し、さらに $\text{unit}(\text{unit } 1, [2, 3, 4, 5,$

14, 15, 16, 17, 18, 19, 25, 26])述語により, 主軸とその周辺部品からなるユニットを導入した。

この主軸台に対しては, 表2に示す63個のエージェントが生成された。本来, 部品数(31)×クラスエージェント数(26)=806個のエージェントが生成される可能性があるところを, 生成条件(base-condition)により63個に絞り込んだことになる。

組立分解プランニングの実験を行った結果, 機械全体の組立/分解, および, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 20, 21, 22, 23, 24, 27, 28, 29, 30, 31のうちの任意の部品の取外し, ないし, 交換の目標に対して適切な手順を生成することができた。また, プラン生成中の任意のタイミングに目標を変更しても, 新しい目標に対して適切な手順を求めることができた。

一方, 部品16, 19, 25, 26については, unit/2述語により定義されたunit1を取外した後に取外し作業が行われた。これらの部品は, unit1を取外さなくても取外し作業ができるので, 遠回りの分解手順を生成してしまったことになる。目標によっては, ユニットの構成を変更する必要がある。

システム動作の安定性を評価するために, β (後ろ向き活性度伝達比率)および γ (外部入力におけるゴールからの入力比率)を0.0~1.0の範囲で0.1刻みで値を変えて実験を行った。その結果, $0.5 < \beta < 1.0$, $0.0 < \gamma < 1.0$ の範囲で適切なプランニングを行うことができ, 十分な動作安定性をもつことがわかった。

図4に組立分解プランニングのシミュレータを示す。これは部品を取外したり, 組付けるようすをアニメーションで表示することができる。目標指示は対象となる部品番号(全体を対象とする場合はall)を入力し, DisassemblyまたはAssemblyのボタンをクリックすることにより行う。実験はSparc 10ワークステーションで, Sicstus Prolog⁽⁹⁾により行った。インスタンスの生成とリンクの設定に約1分, 全体の分解/組立にそれぞれ約30秒程度の計算時間がかかった。

5. 結 言

エージェントの協調作業による機械の組立分解プランニングシステムを開発し, いくつかの例題で実験を行った。このシステムは予期しない目標や状態の変更に対応でき, 指定部品の交換など, 種々の目標に対して, おおむね適切な組立分解手順を生成できたが, 遠回りの手順を生成する場合もあった。目標に応じて適切な組立分解手順を求めるため, クラスエージェントの開発改良を進める必要がある。また, 組立構造デー

タ(fit/5, face/5)をCADデータから直接求める手段を開発し, CADシステムとリンクさせることも課題である。本システムの適用例としては, 機械組立, メンテナンスのための手順指示, 組立分解コスト計算などが考えられる。

本システムのアルゴリズムは, 積み木の世界におけるロボット動作計画⁽⁶⁾にも適用可能である。

付録 エージェント定義に 用いた述語の意味

(1) positioned(P): 部品 P が他の部品に組付けられている。すなわち, P を含み Truth= t のfit/5またはface/5述語が存在する。

(2) assembled(P): P は単一の部品または組立てられたユニットである。

(3) dismounted-unit-member(P): P は単一の部品, または他の部品群から取外されたユニットである。

(4) bolt(P): P はボルトである。すなわち, scによる外周はめあい一つ, moによる外周はめあい一つ以上あり, 内周はめあいはない。

(5) nut(P): P はナットである。すなわち, scによる内周はめあい一つあり, 外周はめあいはない。

(6) fit(Dir, P1, P2): P2の内周部分とP1の外周部分は直接または間接的(いくつかの部品を介して)はめあい関係にある。

(7) face(Dir, P1, P2): P1の左側の面とP2の右側の面は直接またはいくつかの部品をはさんで, 向かい合っている。

文 献

- (1) 北嶋克寛・吉川弘之, 機械の有向接続グラフに基づく対話型組立可能性判定システムの開発, 精密機械, 49-2(1983), 208.
- (2) 伊藤公俊・曾布川拓司, 機械のアセンブリ計画の自動生成(第2報)-システムの実現-, 精密工学会誌, 57-1(1991), 104.
- (3) 内山直樹・荒井栄司・井越昌紀, 機械製品の組立可能性の検証のための分解経路の生成, 機論, 59-560, C(1993), 1300.
- (4) De Mello, L. and Sanderson, A., And/Or Graph Representation of Assembly Plans, IEEE Trans. Robotics Autom., 6-2 (1990), 188.
- (5) 関口博・今村聡・小島俊雄・井上久仁子, 回転機能部品の部品展開に関する研究(第2報)-組立分解手順の生成とその規則化, 精密工学会誌, 53-8(1987), 1183.
- (6) Seow, K. T. and Devanathan, R., Temporal Logic Programming for Assembly Sequence Planning, AI in Engineering, 8-4 (1993), 253.
- (7) Maes, P., The Dynamics of Action Selection, Proceed-

- ings of IJICAI*, (1989), 991.
- (8) 今村聡・榊原寿・堀江優美子・榎本進, ロボットハンドの最適操作列を導くエージェント群の協調動作方式, 機械技術研究所報, 47-6(1993), 227.
- (9) Andersson, J., ほか 6 名, SICStus Prolog User's Manual, *SICS Technical Report, T93 : 01*, (1993), Swedish Institute of Computer Science.
-