

## 組立て知能ロボット ARI のプランニングシステム†

大嶽能久<sup>††</sup> 隅田敏<sup>†††</sup> 水谷博之<sup>††</sup>

産業用ロボットは近年急速に普及が進んだが、主にプログラミングの難しさがより難度の高い複雑な作業への適用の障壁となっている。われわれはこのような状況を改善する将来型ロボット・システムの研究環境としての、次世代組立てロボットのプロトタイプシステム ARI (Assembly Robot with Intelligence) を開発した。今回プラスチック製ブロック玩具の長短2種類のブロック数個を使って適当に組み立てた見本を提示し、その構造を視覚を用いて認識させ、それと同じ構造のものを実際に組み立てさせるという実験を ARI を使って行った。本論文ではこの ARI のために開発したプランニングシステムについて述べる。ARI のプランニングシステムは前段でのステレオ視による組立て見本の構造推定の結果を受けて、その組立て方を計画しロボットの動作手順を計画し、後段の制御系へのロボットプログラムを生成する。このプランニングシステムは2つのモジュールによって構成した。上位のモジュールを作業プランナ、下位のモジュールを動作プランナと呼ぶ。作業プランナは非線形計画階層計画システムとして構成し、ロボットによる実作業のプランニングに適用するために、環境の詳細なモデルとしてのワールドモデルとの融合を行った。動作プランナは作業プランナが生成する作業レベルの命令をロボットによって達成する方法を計画する。その問題解決知識を属性文法と類似した作業モデルとして整理した。

### 1. はじめに

産業用ロボットは近年急速に普及が進み、より難度の高い作業への適用が期待されるようになってきている。そのためには、ロボットをこまごまとした指令やティーチングなしで動かすことができるようになるのが望ましい。その1つの理想形は、完成品のデータを提示するだけでロボットが自ら組立て手順を考えて作業を行う姿であろう。われわれは次世代組立てロボット実験システム ARI (Assembly Robot with Intelligence) をとおしてその可能性を検討してきた。本論文では ARI のために開発したプランニングシステムについて述べる。その主な特徴は、

- (a) 問題解決フェーズにそった自然なモジュール分割と、それぞれのフェーズに応じた計画法。
- (b) 領域独立計画システムと領域依存ワールドモデルとの作業選択機構による融合。
- (c) 作業モデルによる動作計画のための知識表現と、作業の意味的構造を表現する作業の構造木である。

### 2. ARI の構成と機能

ARI は人間の上半身をモデルに構成された (図 1) 次世代知能ロボットの開発に必要な基礎的な技術の研究のためのプロトタイプシステムである<sup>3)</sup>。ARI はポテンシャル法によって自動的に障害物をよけ、視覚フィードバックと力制御で位置ずれを補正するという強力な環境適応能力に支えられて、軽量かつ高自由度のアーム・ハンド系による器用な作業が可能となるように構成されている。

ARI による自動組立て実験をプラスチック製ブロック玩具を使って行った。その内容は、長短2種類のブロック数個を使って適当に組み立てた見本を提示し、その構造を視覚を用いて認識させ<sup>7)</sup>、それと同じ構造のものを実際に組み立てさせるというものである。

### 3. プランニングシステム

ARI のプランニングシステムは前段での組立て見本の構造推定の結果を受けて、その構造をバラ置き状態のブロックを使って組み立てさせるためのロボットの動作手順を計画する。結果は後段の制御系へのロボットプログラムとして出力する<sup>11)</sup>。

ARI のプランニングシステムは問題解決過程を2つのフェーズに分けて解く。第1のフェーズではロボットによる操作の対象となる個々の部品についての状態遷移に関して推論し、目標構造物の組立て手順を決定する。第2のフェーズでは個々の組付け作業をロボッ

† Planning System for an Intelligent Assembly Robot ARI by YOSHIHISA OHTAKE (Research & Development Center, TOSHIBA Corporation), SATOSHI SUMIDA (Manufacturing Engineering Laboratory, TOSHIBA Corporation) and HIROYUKI MIZUTANI (Research & Development Center, TOSHIBA Corporation).

†† (株)東芝総合研究所情報システム研究所

††† (株)東芝生産技術研究所

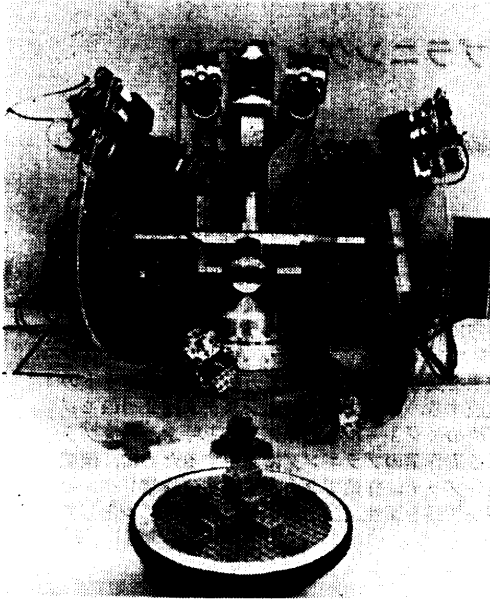


図 1 ARI の全体像  
Fig. 1 An overview of the ARI.

トによってどのようにして実行させるかを推論し、ロボットプログラムまで詳細化する。以下動作モジュールとはアームやハンドやカメラなどの、ロボットを構成する各機能単位のことであり、また動作パラメータとはアームの位置や指の配置および開き幅など、各動作モジュールの各アクチュエータへの制御量を決定するパラメータである。

ARI のプランニングシステムは上記の 2 つの問題解決フェーズに対応する 2 つのモジュールによって構成した。それぞれ上位モジュールを作業プランナ、下位モジュールを動作プランナと呼ぶ。

#### 4. 作業プランナ

このモジュールは非線形階層計画システムとして構成している。階層計画とは大まかな計画からより詳細な計画へと、段階的に計画を進めていく方法である。非線形計画とは Sacerdoti<sup>9)</sup> や Tate<sup>12)</sup> などによって提案され拡張されてきたもので<sup>11), 21)</sup>、必要が生じた場合にのみ作業間に順序関係が導入される。つまり作業間の複数の線形順序をそれらと等価な半順序によって代表させることによって、探索空間を削減するという効率性と、作業の並列性を陽に扱えるといった優位性がある。反面限界も指摘されているが、作業プランナではワールドモデルを導入することによって問題点を解消している。

#### 4.1 作業プランナの基本原則

作業プランナにおける計画作成は、2 段階の処理の繰返しによって行われる。第 1 段階は各々のサブゴールを達成する作業の選択。第 2 段階は作業間の競合の検出と解消である。競合とは順序関係のない作業間の前提条件と効果の間に矛盾が含まれること、つまり半順序と等価な線形順序の集合中に正しくない計画が含まれていることをいう。

##### 4.1.1 作業選択

作業プランナでは作業に関する知識を作業スキーマとパターンスキーマの 2 種類のスキーマで表現している。作業スキーマには作業に関する定義が、

(a) 前提条件: その作業の実行時点で成立していなければならない世界の状態。

(b) 効果: その作業によって引き起こされる世界の状態の変化。

(c) 資源: その作業に要求される資源。

(d) 展開情報: より詳細な要素作業への展開情報。

の各項目について記述されており、他のプランニングシステム同様、作業の前提条件と効果は、追加・削除リスト形式で定義されている。その他、ゴールは一階述語で記述し、計画はオペレータをノードとするネットワークで表現されるのも他と同様である。

一方パターンスキーマには作業スキーマの選択とその具体化の方法が手続的に定義されている。第 1 段階の処理ではこのパターンスキーマを使ってサブゴールを達成する作業の選択をする。その様子を図 2 に示す。

達成しようとするサブゴールの記述を検索パターンとしてパターンスキーマが検索される。次いでそこに定義されている作業スキーマの選択手続きが起動される。この選択手続きは後に述べるワールドモデルにアクセスし、作業スキーマの選択と具体化に必要な情報を検索する。その情報に基づいて、その状況に適した作業スキーマが選択され、同時に作業スキーマに定義されている前提条件・効果中のパラメータを具体化する。

##### 4.1.2 競合検出・解消

第 2 段階の処理は、前記作業の定義の (a), (b) に基づいて競合を検出しそれを解消する、つまりある作業の前提条件を侵害しうる作業の効果の影響を新たな順序関係の導入によって排除することである。ここでの処理は他の非線形計画システムでもほぼ同様であ

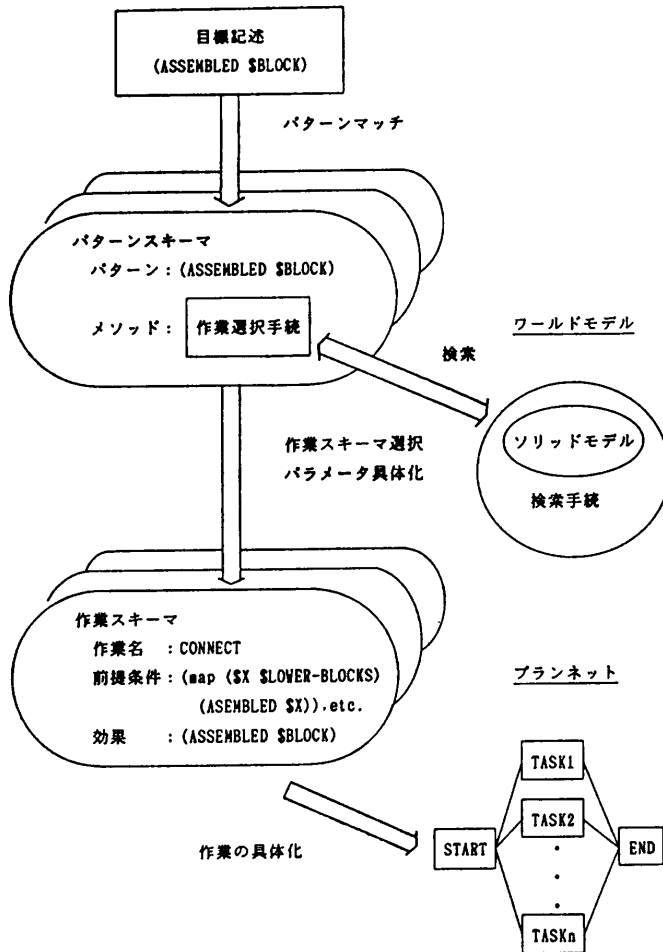


図 2 作業選択機構  
Fig. 2 Task selection mechanism.

る。

さらに作業プラナでは(c)に基づいて、上記の処理の後にも順序関係が付けられなかった潜在的に並列実行可能な作業の集合が全体として要求する資源が、利用可能な資源の範囲を越える場合、資源を各作業に配分するという処理も行っている。

第2段階での処理の様子を図3に示す。図の箱枠は個々の作業に対応するオペレータである。(a)図中の箱枠の左肩は前提条件、右肩は効果、～は否定、+は世界の状態記述への追加、-は削除、矢印はオペレータ間の順序関係をそれぞれ表している。

最初は各オペレータ間には順序拘束がなにも設定されていない状態から処理を開始し、ある作業の前提条件を侵害しうる効果を生成する作業の有無を調べる。例えば(a)図で TASK 3 の実行が TASK 1 に先行すると、TASK 3 の p1 に関する効果が TASK 1 の p1 についての前提条件を侵害してしまうので、点線

矢印の順序が導入されている。さらに(b)図では、TASK 1 もしくは TASK 3 と TASK 4 の間の、資源 1 に関する競合を解消するために点線矢印の順序が導入されている。

以上の第2段階が完了すると、作業の記述(d)の展開情報に従って作業をより詳細なサブゴールに展開し、計画を一段詳細化して第1段階に戻る。作業の記述に展開情報が含まれていなければ作業プラナでの計画はそこで完了する。

#### 4.2 世界の状態表現

以上が作業プラナにおける問題解決の基本原理である。しかしロボットによる実作業のプランニングへの適用を試みるに際して幾つかの問題点が明らかになった。その1つは世界表現の問題である。

プランニングの場合、世界の状態は一階述語論理式で表現されるのが一般的である。作業プラナでもまた、前段で推定された組立て目標となる構造物のデータを連言形式の一階述語論理式に変換し、達成すべき個々のサブゴールを原子論理式によって表現する。

一階述語論理式による世界の状態表現には、その記述形式としての一般性による領域独立性や、透明性と簡潔性の高い処理が可能であるなどの良い性質がある反面、ロボットによる実作業の計画には、対象世界の記述力

が弱いという問題点が発生する。そこで作業プラナでは、世界の状態の詳細データとしてのワールドモデルを導入することによって、一階述語論理式の記述力の弱さを補っている。

ワールドモデルはソリッドモデルをベースにしている。ソリッドモデルは物理的実体の操作を対象とする問題領域での世界の記述に適したモデルである。その基本データは物体の形状や位置に関する情報を数学的なモデルに基づいて表現したものである。作業プラナはロボットの作業計画に必要な対象世界の空間的・構造的性質の扱いをこのソリッドモデルに基づいて行う。

反面ソリッドモデルは一階述語論理式に比べて表現の一般性や処理の透明性が低いので、処理をソリッドモデルと密に結合することは、プランニングシステムの領域独立性を失ってしまう結果となる。

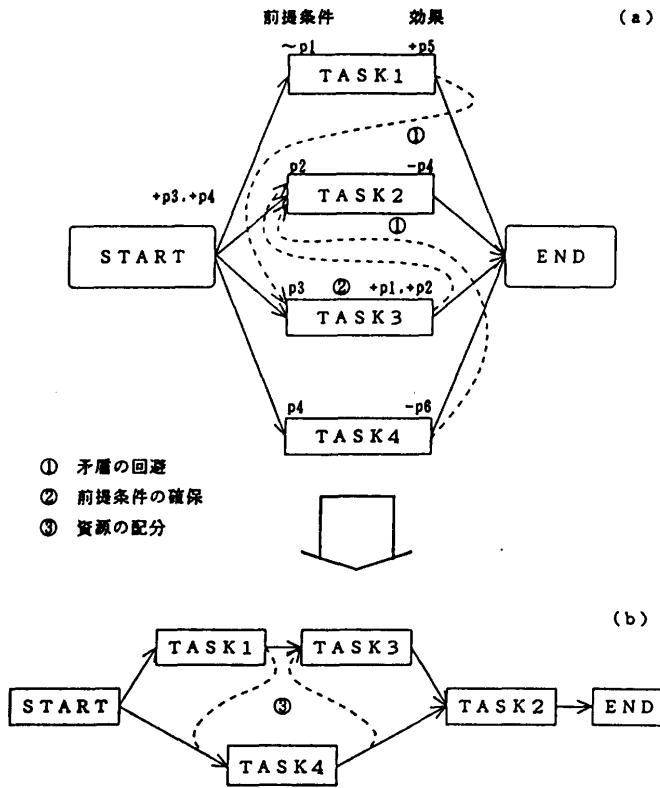


図 3 競合の検出・解消  
Fig. 3 Conflict detection and resolution.

4.3 文脈依存条件

非線形計画法は効率的である反面、限界も指摘されており、ロボットによる実作業のプランニングへの適用でもこの点が問題となった。

非線形計画法では各作業の定義は文脈自由形式で定義される。つまりある作業の定義はその作業単独で局所的に決定可能なものに限られ、作業はあらゆる状況で同じ効果を生成するものとするという仮定がある。これを局所性仮定 (locality assumption) と呼ぶ。

この仮定は競合の検出・解消を容易にし、探索空間を削減する反面、共同効果 (synergy effect)<sup>2)</sup> 等の、作業の文脈に依存した条件、あるいはある文脈の結果としての状況に依存した条件の扱いを困難にしている。

Pednault<sup>9)</sup> は、作業の実行に伴う普通的前提条件である主要前提条件 (primary precondition) と、作業が特定の効果を生成する文脈を定義する 2 次的前提条件 (secondary precondition) という概念を導入し、ある文脈とそのもとで生成されることが期待される効果とを列挙し記述することで、従来の手法の基本的な性質を壊すことなく文脈を扱う方法を提案している。し

(a) かし問題を限定しない限りそのような文脈を、一般的にありうるすべての可能性について列挙することは実現困難である。

一方局所性仮定を取り去った、一般化された競合の概念の定式化も成されているが<sup>4)</sup>、同時にその競合の検出の一般的な方法は、複雑で非実用的であることを指摘している。

両者とも非線形計画法において文脈依存条件を実用的に扱うためには、何らかの探索空間の限定が必要であることを示唆しており、特に後者では領域知識の利用の必要性を指摘している。

4.4 領域知識の利用

ブロック組立ての場合、個々のブロックを目標位置の上方近傍から鉛直下方に加重しながらワーク (組立て途中の構造物) に組み付けてゆくのが標準的な作業方法である。実行不可能でない限り ARI はこの作業方法で組立てを進める。

この作業の前提条件はブロックの性質から、

(1) 各ブロックについて、目標状態において隣接下方に位置するべきブロックは、そのブロックの組付け作業に先行して組付けを完了していなければならない。

(2) 組付けの際の加重に対して安定な把持位置が存在していなければならない。

である。これらの条件はいずれも非線形計画法の範囲内では処理できない。その理由は、

(a) 作業の前提条件は初期状態の記述と他の作業の効果とによってのみ評価されるものである。したがって目標状態の記述を含んだ条件は前提条件として評価することができない。

(b) 組付け加重に対して安定という条件は、その作業の実行時点でのワークの組上り状況に依存する状況依存条件、言い換えればその作業の実行以前の作業の文脈に依存する文脈依存条件である。

ARI のワールドモデルにはロボットの作業環境を表現しているソリッドモデルと、それに基づいてブロック間の隣接関係やブロックの安定性などの情報を抽出する手続きが含まれている。したがって前提条件の評価はワールドモデルにおける目標状態での世界記述に基づいて、条件(1)は具体的なブロック名による記述に置き換え、条件(2)の評価は条件(1)より、作業

台から対象ブロックまで安定位置を隣接上方のブロックに伝播させることによって評価することができる。

これらの処理は作業選択機構からワールドモデルに問い合わせる形で行われる。もし条件(2)の評価で安定位置がないという評価結果をワールドモデルが返したならば、作業選択機構は標準作業以外の作業、例えば部分組立作業を選択する。

この作業は不安定ブロックをその隣接下方あるいは隣接上方の安定ブロックに結合し、それらをサブアセンブリとしてワークに組み付ける作業である。

以上のように作業プラナではワールドモデルを対象世界についての一領域知識ベースとし、作業選択機構をそのインタフェースとすることによってプランニングシステムの領域独立性を保持すると同時に、対象世界についての詳細な情報を利用可能にし、文脈依存条件の処理に関する困難も、ワールドモデルの参照によって解決している。

## 5. 動作プラナ

動作プラナは作業プラナから与えられる個々の作業レベルの命令を現実のロボットによって達成する方法を計画して、各動作モジュールへの動作指令からなるロボットプログラムを生成する。

### 5.1 作業モデル

言語処理系という観点から見ると動作プラナは一種のコンパイラである。実際その基本的な動作原理は属性文法<sup>9),10)</sup>と類似している。そこで動作プラナにおける問題解決知識を、ロボットプログラムの言語モデルに相当するものとして、作業モデルと呼んでいる。以下、属性文法と作業モデルとの対比によって、動作プラナの機能と性質を明らかにする。

プログラミング言語は大まかに構文 (syntax) と意味 (semantics) とからなる。属性文法は Knuth によって提案されたもので、意味規則によって構文と意味とを一体化して記述できる。構文についてはその可能な構文木の形を記述するための構文規則 (通常は文脈自由文法) によって定義される。木中の節点上には属性が設定され、意味規則はその値の決定方法の定義である。

属性文法 = 構文規則 (文脈自由文法)  
+ 属性 + 意味規則

作業モデルも同様の構造をしている。

(a) 展開規則: 構文規則に相当するもので、作業間に存在する構成関係を定義する。つまりある作業を

より詳細な作業に展開するための規則である。ただし、作業の実行順序関係 (半順序関係) の定義も含まれている。終端記号は各動作モジュールのプリミティブな動作である。

(b) 意味規則: 属性に相当するロボットの動作パラメータの決定方法である。ただし、属性文法のように合成属性、相統属性といった分類はしない。

ARI は前記のようにアームやカメラなどの、複数の動作モジュールによって構成されているので、作業の実行効率の観点から、これらの動作モジュールを無用な待ちが生じないように協調させるようなプログラムを生成しなければならない。つまり並列動作が可能な動作モジュールに対する動作命令は並列に実行されるように、展開規則では動作命令の半順序を規定している。したがって順序関係が規定されていない動作命令は後段のロボットコントローラによって非同期並列に実行される。

属性文法はオブジェクト指向プログラミングと強い関連を持つことが指摘されている。同様の理由から動作プラナもオブジェクト指向パラダイムに基づいて構築されている。終端記号を含めて各作業をオブジェクトとして表現し (作業オブジェクトと呼ぶ) プランニングはメッセージパッシングによって進められる。

### 5.2 作業の構造木の生成

属性文法では、まず生成規則によって構文木の生成を行い、その上で属性計算を行う。動作プラナもまず展開規則によって作業の構成関係を表す木構造データを生成し、その上で必要な動作パラメータの決定を行う。木構造データの生成は、

(1) 作業プラナからの作業命令に対応する作業オブジェクトを初期記号とし、それに対する作業展開メッセージによって開始される。

(2) 作業展開メッセージを受けた作業オブジェクトはまずそのインスタンスを生成し、メッセージを送ってきたインスタンス (初期記号の場合はシステム) を上位オブジェクトとして記録する。このインスタンスは具体化された動作パラメータの記憶場所となる。

(3) 次にそのインスタンスから展開規則に従って、逐次下位の作業オブジェクトに作業展開メッセージを伝播させ、生成されたインスタンスを下位のオブジェクトとして記録する。この操作は展開規則を持たない作業オブジェクト、つまりプリミティブ動作の作業オブジェクトに達するまで順次繰り返される。

以上の結果、作業オブジェクトのインスタンスから

なる、作業の構造を表す一種の構文木が生成される。これを作業の構造木と呼ぶ。例として ARI で使われている作業 ATTACH (1つのブロックをワークに組み付ける作業) についての作業の構造木の一部を図4に示す。

図4における弧は抽象作業あるいはプリミティブ動作間の順序関係を表している。1つの作業オブジェクトから展開された作業の集合は先にも述べたように既に半順序化されている。異なった作業オブジェクトから展開された作業間での順序関係は、原則的にそれらの上位作業間での順序関係に従うものとする。しかしそれでは下位作業間に不必要な順序関係を導入することがある。このような過剰な順序付けが成されることを避けるために、展開規則には下位作業の目的を規定することができる。例えば「カメラを動かす」(MOVE-CAMERA)の目的は「視覚補正」(VISUAL-FEEDBACK)の準備をすることであるといった、動作の目的を記述することができる。目的を持った要素作業の順序付けは原則的な順序付けに優先して行われ、過剰な順序拘束の付加による無用な待ちの発生を防いでいる。

### 5.3 動作パラメータの評価

動作パラメータの評価は上記のようにして作られた作業の構造木の上で行われる。これは属性文法における属性評価に相当するものであるが、その処理の仕方や性質は幾つかの点で大きく異なる。

まず属性文法ではルール形式の意味規則に従って属性評価が行われるが、動作プラナでの動作パラメータの決定方法は、作業オブジェクトのインスタンスメソッドとして手続的に定義されている。したがって、属性文法のような記述の明快さはない反面、属性文法では記述できない柔軟な処理を実現している。動作パラメータの決定は以下のような流れで処理される。

(1) 作業の構造木の生成時と同じく、作業プラナからのメッセージパッシングによって開始される。異なる点は、今回は初期記号に対応する作業オブジェクトのインスタンスに対して送られる点である。

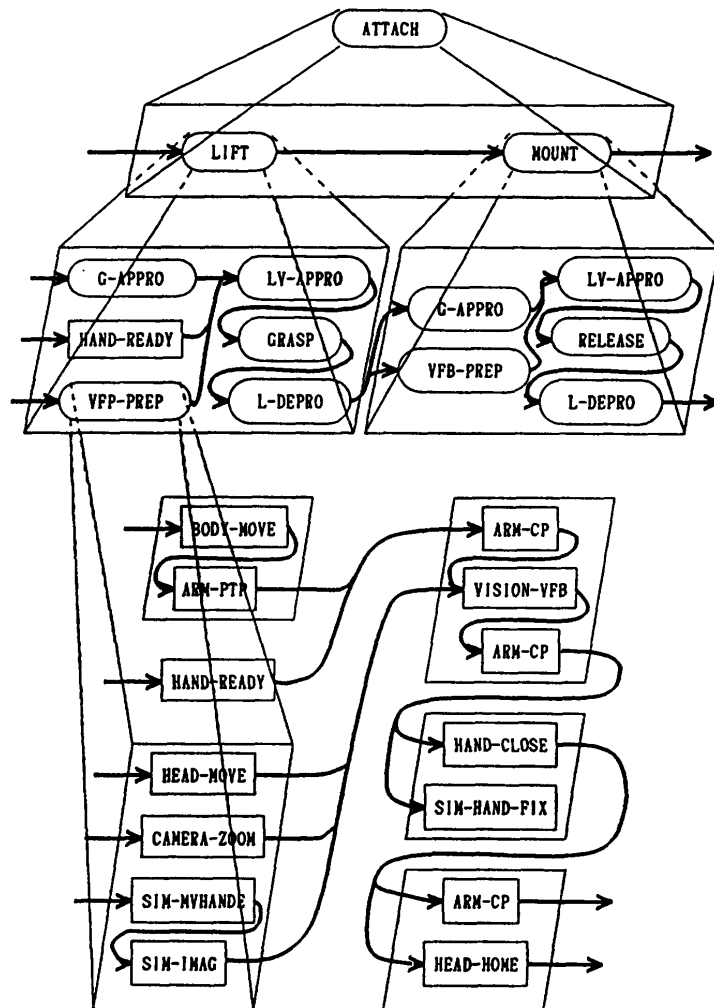


図4 作業 ATTACH についての作業の構造木

Fig. 4 Part of a task structure tree for ATTACH.

(2) メッセージを受けた作業オブジェクトは、メソッドを起動して動作パラメータの決定に必要な処理を行う。処理が定義されていなければ、そのまま下位のオブジェクトにメッセージを伝播させる。

(3) 決定した動作パラメータ等の必要なデータはインスタンス変数に記録しておく。

属性文法の相続属性の評価に相当する処理はインスタンス変数についてのインヘリタンスに基づいて行われる。動作パラメータには1つのオブジェクトによる処理で局所的に決定可能なものや、複数のオブジェクトによる処理結果から決められるもの、さらに他の動作パラメータに依存して決定されるものなどがある。そのため属性文法の属性評価のような一様な制御の流れでは処理できないものがある。動作プラナでは動作パラメータは、メッセージパッシングによって制御を

作業の構造木上で上下させながら、上位のオブジェクトから段階的に決定されていく。

例えばブロックの把持位置の決定では、作業時の組付け加重に耐えられる安定な位置であり、かつハンドが既に組み付けられた他のブロックと干渉しない位置を選択しなければならない。これは複数のオブジェクトによる処理の結果から決められる動作パラメータで、しかもその値に依存して決まる多くの動作パラメータが存在する。その値の決定は、まず木中の把持位置の決定に関与するすべての作業（例えばつかみ上げる作業やワークに組み付ける作業など）の上位の作業オブジェクトによって、対象物のすべての把持位置の候補がワールドモデルから取り出される。次にそのオブジェクトは先の各作業オブジェクトに対してメッセージパッシングによってその中から可能なものを選択させる。もとのオブジェクトはその結果を集計して適切な値を決定し、下位のオブジェクトにインヘリタンスによって伝達する。そしてアームの誘導位置や指の配置、そしてカメラの目標位置などがその値に従って、下位のオブジェクトで決定される。

属性文法では一度生成された構文木は不変であるが、動作プラナでは作業の構造木も動的に変化する。つまり作業のモデルは最長手順を生成し、不必要な枝の枝刈りを動作パラメータの決定と同時に行う。例えば上記の例で、共通の把持位置が得られない場合、作業の途中で把持位置を変更する動作が必要である。作業の構造木はこの動作を想定して生成されるが、動作パラメータの決定で把持位置の変更が不要と分かれば、その動作に対応する枝を作業の構造木から刈り取る。また属性文法では大域変数の参照は行われず、必要なデータ類は相続属性として受け渡されるが、動作プラナでは作業プラナ同様、動作パラメータの決定の際には一種の大域変数であるワールドモデルを参照する。

属性文法とのもう1つの重要な違いは、属性文法は副作用を含まない純粋な関数型プログラミング言語で、値を格納する場所としての変数という概念を持たない。つまり外部から入力データを受け、構文木上で属性計算を行い、その結果を外部に返すのが目的である。一方動作プラナは動作パラメータの値で肉付けされた作業の構造木を作成するのが目的である。

#### 5.4 作業の意味的構造

先にも述べたように作業の構造木は構文木に相当するものである。この構文木は言語理論においては文の

意味を理解する上で最も基本となる重要な情報である。同様に作業の構造木も、一連のプリミティブ動作によって実行される作業の意味的構造を表す情報として理解することができる。つまり個々には無性格なプリミティブ動作に対して、この情報を使って意味付けを与えることができる。このような情報には以下のような有用な利用方法が考えられる。

- (1) 作業の実行段階に応じたセンサ情報の解釈と、それに基づいた状況の理解能力を与える。
- (2) エラーを検出した場合、エラーの原因の分析や、エラー処理の方法を推論する際に利用する。

## 6. 実行例

ある例についての作業計画の作成結果のシミュレーションのようすを図5に示す。図中の各直方体はブロックを、三角柱はハンドの指を表している。②③⑥では指が既に組み付けられている他のブロックと干渉しない把持位置が選ばれていることが分かる。④は組付け加重に対して安定な位置が存在しないブロックと、それに隣接し安定位置が存在するブロックとを両手を使って空中で部分組立てする作業が作業選択機構によって選択され実行されているようすである。⑤ではワークへの組付けのための把持位置に持ち変えられていることが分かる。一連の把持位置は組付け時の把持位置を基準にして、左右のハンドの指が干渉しないよ

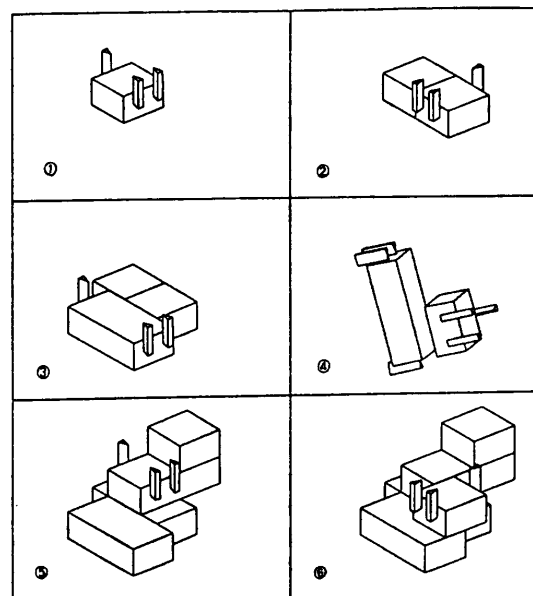


図5 作業計画の生成例

Fig. 5 An example of assembly plans generated by the ARI planner.

うに動作プランがスケジューリングする。

ただし干渉チェックには一般的な干渉計算法ではなく簡略化した方法を使って処理時間を短縮している。

この例で生成された動作手順は約 230 個のプリミティブ動作からなり、その生成には 240 CPU タイムを要している。作業プランおよび動作プランはともに AS 3000 (TOSHIBA 32-bit EWS) 上に KCL (Kyoto Common Lisp) によってインプリメントした。

## 7. おわりに

ロボットによる組立て作業のような、物理的実体の操作を扱うプランニング技術の開発<sup>5)</sup>を上記のようなアプローチで行い、実際に ARI による自動組立て実験をプラスチック製ブロック玩具を使って行った。

しかしその問題解決能力はまだ十分に一般的なものではない。本研究では未解決または未着手な問題の中で、特に重要性を認識した研究課題を列挙すると、

(a) 現実世界環境の性質を一般的かつより詳細に扱うことのできるワールドモデル。

(b) 実行エラーや作業の副次効果などによって生じる予想外の環境の変化に対する動的な計画修正などがあげられる。

(a)に関して、ARI の現在のワールドモデルはソリッドモデルを除いて、ブロック組立てのための領域知識として特化した部分が多い。そのため ARI の実行可能な作業の範囲は現状では限定されたものである。より一般性の高いワールドモデルの構成法は未解決な問題である。

しかし、ワールドモデルによって現実世界の事象を扱うことには技術的、コスト的に限界がある。したがって(b)は(a)同様あるいはそれ以上に重要な研究課題であると考えている。ARI ではそのために作業の構造木を用意しているが、それを使った動的計画修正は現在も未着手である。またこれは視覚・触覚を中心とする環境認識能力の研究に期待するところが大きい。

## 参考文献

- 1) 安部憲広: 作業プランニング, 日本ロボット学会誌, Vol. 5, No. 6, pp. 480-486 (1987).
- 2) Chapman, D.: Planning for Conjunctive Goals, *Artif. Intell.*, Vol. 32, pp. 333-377 (1987).
- 3) Fujita, N.: Assembly of Blocks by Autonomous Assembly Robot with Intelligence (ARI), *Annals CIRP*, Vol. 37, No. 1, pp. 33-36 (1988).

- 4) Hertzberg, J. and Horz, A.: Towards a Theory of Conflict Detection and Resolution in Non-linear Plans, *Proc. IJCAI-89*, pp. 937-942 (1989).
- 5) Inoue, H.: Building a Bridge between AI and Robotics, *Proc. IJCAI-85*, pp. 1231-1237 (1985).
- 6) 片山卓也: 属性文法—構造的かつ関数的計算モデル—, 情報処理, Vol. 29, No. 2, pp. 146-154 (1988).
- 7) Mizoguchi, H.: Visual Processing in a Robot Which Assembles LEGO Blocks, *Proc. of ROVISEC-7*, pp. 185-195 (1988).
- 8) Pednault, E. P. D.: Extending Conventional Planning Techniques to Handle Actions with Context-Dependent Effects, *Proc. AAAI-88*, pp. 55-59 (1988).
- 9) Sacerdoti, E. D.: The Nonlinear Nature of Plans, *Proc. IJCAI-75*, pp. 206-214 (1975).
- 10) 佐々正孝: 属性文法, コンピュータソフトウェア, Vol. 3, No. 4 pp. 73-91 (1986).
- 11) Sumida, S.: A Planning System for an Automatic Assembly Robot, *Proc. of 18th ISIR*, pp. 169-178 (1988).
- 12) Tate, A.: Generating Project Networks, *Proc. IJCAI-77*, pp. 888-893 (1977).

(平成元年 4 月 3 日受付)

(平成 2 年 1 月 16 日採録)



大嶽 能久 (正会員)

昭和 33 年生. 昭和 59 年京都大学大学院工学研究科精密工学専攻修士課程修了. 同年, 東京芝浦電気(株)入社. 現在(株)東芝・総合研究所情報システム研究所所属. 平成元年 4 月より(財)新世代コンピュータ技術開発機構に出向中. 並列プログラミングの研究に従事. 人工知能学会会員.



隅田 敏 (正会員)

昭和 32 年生. 昭和 57 年大阪大学大学院工学研究科精密工学専攻修士課程修了. 同年, 東京芝浦電気(株)入社. 現在(株)東芝・生産技術研究所研究主務. 主として, CAD/CAM, ロボット制御ソフトウェアの開発に従事. 計算機援用の情報生成システム, マンマシンインタフェースに興味を持つ.



**水谷 博之 (正会員)**

昭和 22 年生。昭和 48 年名古屋大学理学部数学科修士課程修了。同年、東京芝浦電気(株)入社。現在(株)東芝・総合研究所情報システム研究所主任研究員。システム技術、知識処理技術の研究に従事。人工知能学会、ME 学会、電気学会各会員。