人　工　知　能 98−10
（1995. 1. 18）

積木世界での環境変化に対するプランの効率的修正

藤田 智之，阪本 利幸，小川 均
立命館大学理工学部情報学科

従来の計画では環境変化によるオペレータの適用失敗には再計画による対応しかなかった．しかし，変更範囲が計画の規模に比して小さな場合には非効率的な処理となる．

そこで本研究では，環境変化の影響検出にＴＭＳを，また，修正箇所及び方法決定に問題設定システムを新たに導入することで，失敗した既存計画の一部に適当な修正を施し変化に対応させる手法を実現し，また，計画適用過程のマニピュレータの動作をグラフィック・シミュレートするシステムを実現した．さらに，計画する動作の抽象度に応じたオペレータ定義や処理レベルの階層化により修正範囲の限定を実現し，最小限度の修正による環境変化への対応を実現した．

Efficiently Modification Method for Unexpected Environment

Tomoyuki Fujita, Toshiyuki Sakamoto, Hitoshi Ogawa

Dept. of Information Engineering,
Faculty of Science & Engineering,
Ritsumeikan Univ, Kusatsu 525-77, Japan.

So far as the ability of planner only, once the validity of plan is lost under unexpected condition, the only way to recover from failure is Reconstruction. Then it becomes very inefficient way, if the part which need to be changed is smaller than the entire plan.

The system proposed in here can modify the used-plan in changing least part of plan-network to keep its validity under new situation.

It consists of three major systems: Operator Modifier(including non-linear planner & TMS & Organizer),Path Planner and Graphic Simulator. And with the hiearchical definition of operators, this system can make the range of modification to minimize.

1 はじめに

計画やスケジュールを作成するという古くからの大きな課題に対しSTRIPS [Fikes 71] をはじめNOAH [Sacerdoti 75],Abstrips [Sacerdoti 74],SIPE [Wilkins 84] などのプランニングシステムが提案され, 世界の初期状態から目標状態を生成するための妥当なオペレータ適用順序の自動的決定を可能としている. しかし, ある初期状態・目標状態に対しては妥当なオペレータ列も与えられる初期状態が異なれば適用できなくなる場合があり, 変化した世界状態からのオペレータ列を再計画しなければならない. これは, 新たに生成されるオペレータ数に比して, 必要とされる変更範囲がごく小さい場合などには, 非効率的と思われる. また, 世界状態の記述に必要とされる状態表記種の増加に伴い, 環境変化とそれに伴うオペレータ適用失敗の可能性が増加し, さらに, 計画の大規模化により計画全体の再構築の非効率度合は増加すると思われる. そこで, オペレータ適用失敗時点の世界状態をもとに適用失敗した計画の一部をうまく修正することでプランを実行可能なものにできれば非常に有効である.

さらに, 理論上は妥当なはずの計画もマニピュレータにより実行しようとした場合には, プランナの認識しえない経路上の障害によって失敗してしまう可能性も存在する. このとき, 障害物との衝突を認識した時点で経路制御に関してはマニピュレータに自動修正を行なわせ, 回復不能な場合にのみ計画修正機構へフィードバックを行なうと効率であると考える.

また, 計画を構成するオペレータや状態表記の表現についても, 計画をより抽象度の高い計画レベルから実世界に近いレベルまで階層的に定義することで修正範囲を最小化することが可能となる.

そこで本研究では, TMS(Truth Maintenance System)[Doyle 79]・プランナ・問題設定システムの三つのシステムからなるオペレータ修正部とマニピュレータの移動経路計画部および障害物－マニピュレータ間の衝突検知機構としてのグラフィック・シミュレータ部をそれぞれ実現, 統合することにより失敗計画の効率的修正手法を提案する.

2 システム構成

本修正システムは図 1 左に示すように, 大きくはオペレータ列修正部 [藤田 94], 移動経路決定部, グラフィックシミュレーションシステムに分けられており, さらに, オペレータ修正部は

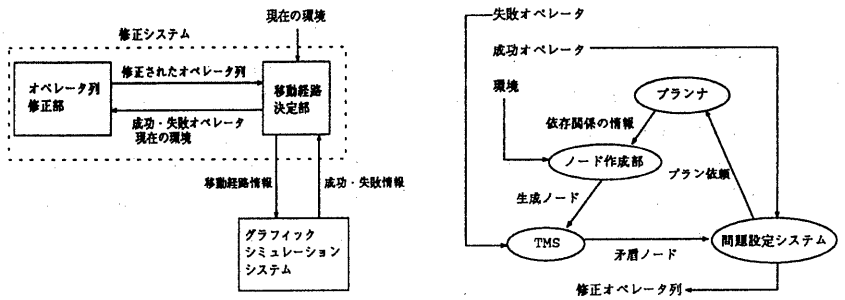

図 1: システム構成 (左) とオペレータ修正部 (右)

図１右のように，問題設定システム，プランナ，ＴＭＳ，ノード作成部の４つのシステムで構成されている．

2.1 オペレータ修正部

- プランナ

本システムにおけるプランナの計画生成手法はＮＯＡＨのそれに準じたものである．ＮＯＡＨの計画生成手法とは，目標に対する非線形な計画をオペレータ間の相互干渉を考慮した順序付けと冗長要素の除去の繰り返しにより生成していくため，オペレータ適用時の前提条件と適用後に成立する状態表記およびオペレータとの依存関係に関する情報の獲得が容易であるばかりでなく，計画中のオペレータ同士の相互干渉が既に考慮済であることから，修正を行なう際の部分計画に伴うオペレータの追加・削除を行なう上で都合が良い．また，オペレータの適用は手段-目標解析で行われる．すなわち，目標状態とされるある状態表記を生成可能なオペレータ (追加リスト中にその状態表記を含んだもの) を適用し，その前提条件の成立をチェック，初期状態において成立していなければその状態表記を副目標として同様のオペレータ適用過程を繰り返すものである．本システムではさらに，オペレータ表現を階層的に定義することで修正対象範囲を限定した．すなわち，レベル１・レベル２のオペレータそれぞれが特有の状態表記に対して適用されるため，修正に必要な最小限度のオペレータ操作による修正が可能となる．例えば図２で示される初期状態，目標状態に対する

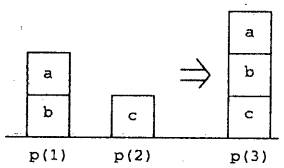

図 2: 問題例

計画はつぎのように計画される．まずレベル１において

stack(a,p(4))-stack(c,p(3))-stack(b,c)-stack(a,b)

が計画される．そのネットワークの一部を図３に示す．つぎにレベル２で図４に示すよう

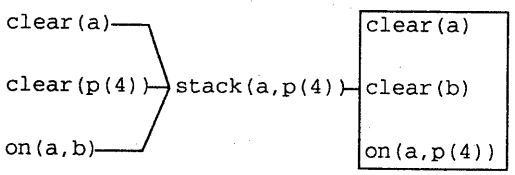

図 3: レベル１での計画 (一部)

なネットワークに展開される．得られるオペレータ列は

pickup(a,b)-puton(a,p(4))-pickup(c,p(2))-puton(c,p(3))-
-pickup(b,p(1))-puton(b,c)-pickup(a,p(4))-puton(a,b)

である．プランナで生成されたオペレータに関して前提条件・追加リストとなる状態表記の情報がノード作成部に送られる．

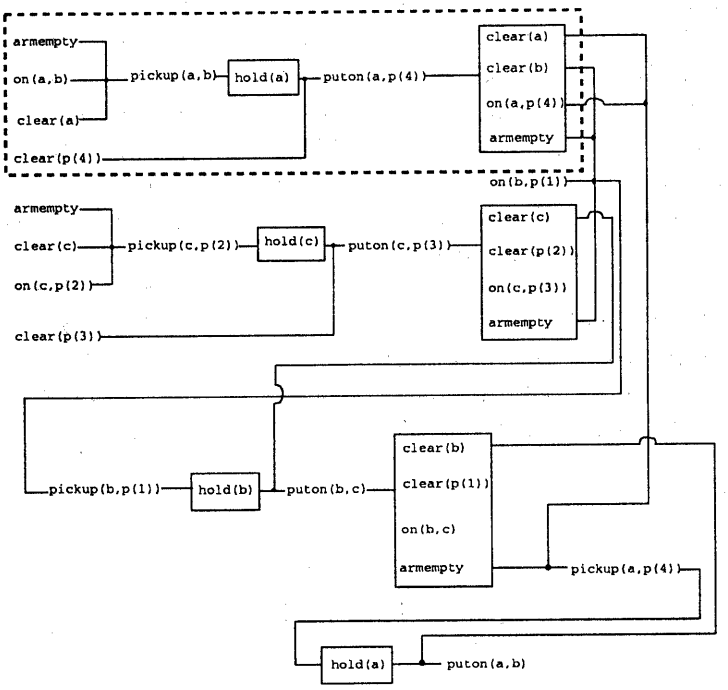

図 4: レベル 2 のネットワーク

- **ノード作成部**

ノード作成部ではプランナから得られるオペレータ・状態表記間の依存関係をＴＭＳが取り扱うノード形式のデータに変更する．また，オペレータ適用失敗が起った場合には，移動経路決定部より伝えられた現在の環境を基にオペレータ適用失敗時における状態表記の成立・不成立を認識し新規ノードの作成を行う．ノードのフォーマットは

(< ノード識別子 > < 信念 > sl([in_list],[out_list]))

のように表される．この形式による理由付けはSL型理由付けと呼ばれる．ここで信念 (belief) は計画中の各オペレータと状態表記を表す．また，in_list には，対象ノードの成立を支持するノードのうち状態が真であるノードの識別子が，また一方 out_list には状態が偽でなければならないノードの識別子がそれぞれ記述される．従って，図 4 におけるオペレータ pickup(b,p(1)) とその適用前後の状態表記は次のように記述される．

node(o(4),pickup(b,p(1)),[sl([c(29),c(18),c(7)],[])])

node(c(7),on(b,p(1)),[sl([],[])])
node(c(18),clear(b),[sl([o(9),o(0)],[])])
node(c(29),armempty,[sl([o(7)],[])])
node(c(23),hold(b),[sl([o(4)],[])])

オペレータも状態表記も他のノードが真となることによって成立するため，通常 out_list は ϕ となるが，競合解消によってあるオペレータを仮説として選択したのならば，競合したオペレータに対し，それぞれ新たなノードを作成し，互いのノード番号を out_list 中に記述する．例えばブロックを目標位置以外の場所に一時的に置く場合の置き場所に関して，このような状況が発生する．また，前記の c(7) ノードに見られるように in_list,out_list が共に ϕ であるようなノードは前提 (premise) と呼ばれ，計画の初期状態を示す各ノードがそれに相当する．

- TMS

TMSは計画の修正を行う際に任意のオペレータまたは状態表記のネットワークにおける影響範囲の認識に用いる．その利点は，ノードと呼ばれる構造中の理由付けによる依存関係の表現とこの依存関係に基づく論理的整合性の維持機能にある．本システムにおいては，ノード作成部で生成されたノードを基に各ノードの状態を決定していく．その際，SL型理由付け中の in_list 中の全ノードの状態が真 (in) であり，かつ out_list 中の全ノードの状態が偽 (out) である場合のみ対象ノードの状態は真となり，これ以外の場合は偽となる．しかし，同一ノードに対する理由づけは複数持つことが許され，これら理由づけのうち少なくとも一つ妥当な (真となる) 理由づけが存在すればノードは真となるものとする．また前提ノードの状態は恒に真である．TMSは，あるノードの状態が変化した際，依存関係指向型後戻り法による自動的な状態変化でネットワークの論理的整合性を保たせようとするが，この処理によってノード状態の整合性が得られない場合，矛盾発生となり，依存ネットワークをもとに矛盾原因の推論を行う．適用失敗したオペレータとネットワークの矛盾原因が検知され，現在の環境において影響を受けるオペレータが認識されると，そのノードを問題設定システムに伝える．

- **問題設定システム**

問題設定システムは，矛盾原因ノードと現在の環境下で適用失敗するオペレータノードの情報をTMSより入手し修正手段を決定する．修正手段を新規オペレータの追加に決定した場合，適用失敗オペレータの追加リスト中において他のオペレータに影響を与えるノードを目標状態とし，プランナに部分計画を行わせる．また，修正計画が現在の環境において無矛盾であることが確認された (矛盾が検出されなかった) 場合，修正後の全オペレータ中で既に適用成功したオペレータを除いたオペレータ列を修正オペレータ列として経路計画部へ送る．

2.2 移動経路決定部

移動経路決定部は，プランナが生成した計画中の各オペレータが現在認識されている環境下において適用可能であるかを逐次判定．適用可能と判断され，移動経路を決定されたオペレータについてその実行をシミュレータに伝達する．移動経路決定部における対象世界の座標定義は図 5 のようになっている．

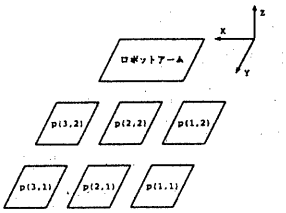

図 5: 移動経路決定部における座標系

プランナにおいて p(i),i=1,2,3,…の型で識別されていたテーブル上の各位置は図 6 に示されるような実座標中の適当な p(x,y) の位置に割り当てられる．移動経路決定部における移動経路は，基本的に z 座標を一定とし，x 軸に沿った移動に次いで y 軸沿いの移動を行うことで対象地点座標への到達可能性を確認する．また，障害物の存在を移動経路上に認識した場合の戦略は次の二通りである．

1. マニピュレータの長さ限界の範囲内において，障害の上を通過可能であれば，移動軌道の z 座標を増加させ，x-z 平面もしくは y-z 平面内での衝突回避運動を行う．
2. 障害物の上をマニピュレータの物理的制約により通過不可能である場合，x-y 平面内での衝突回避運動を行う．その際，x 軸方向への移動に対して，障害物との衝突がある場合，同一 y 座標上に障害が存在しなくなるまで，y 座標を減じることで x 軸沿いの移動が可能な地点まで移動する．

このようにしてオペレータ列の先頭より順次マニピュレータ移動経路を決定していく．また，ここでマニピュレータの手先が障害物を回避することで対象物体を把持・運搬が可能となるような，いかなる経路も存在しない場合，オペレータの適用を不可能と判断し，その時点までで適用成功したオペレータと適用失敗したオペレータ，及び，そのオペレータ列を適用しようとしていた世界の現在の初期状態をオペレータ修正部に伝え，修正オペレータ列の到着を待つ．移動経路を伝達する命令は次のフォーマットで記述される．

act(移動経路変数 1 , 移動経路変数 2 ,
行動, ブロック名, 移動経路上で最も高いブロックの高さ,
目標位置の x 座標, 目標位置の y 座標, 目標位置の z 座標)

ここで，移動経路変数とはマニピュレータが辿る経路および障害物回避の戦略のパタンを示すものである．例えば，

act(1,a,get,a,1,2,1,2)

は座標位置 (2,1,2) にあるブロック a をアームの Y 座標を変えることなしに取りにいくという動作を表わしている．

2.3 グラフィック・シミュレーション部

移動経路決定部において適用を承認されたオペレータをその経路移動計画に基づいて描画・シミュレートする機能と衝突の検知機能を持つ．オペレータ適用シミュレーション過程において，移動経路決定時点では認識されていなかった障害物とマニピュレータとの衝突が検知された場合，障害物の存在座標とその高さを

position(obstacle(番号), ブロックのX座標, ブロックのY座標, ブロックのZ座標)

のように移動経路決定部に伝える．例えば

position(obstacle(0),2,1,1)

は座標(2,1,1)に除去できない高さ1の障害物が置いてあることを示している．ただし，本システムにおいては，障害物形状は全て直方体として取り扱っている．また本稿におけるグラフィックライブラリは，本研究室の卒業研究成果[児玉 92]を利用している．

3 計画修正手順

本修正システムにおける修正は以下の手順により実現される．

1. ある初期状態・目標状態に対して生成されたオペレータ列に対し，それらが現在の対象世界において適用可能となるマニピュレータ移動経路が存在するか移動経路決定部が順次オペレータをチェックし，経路が確定するとシミュレータにオペレータの適用実行を命令する．オペレータ適用失敗が検知されるまで順次適用を繰り返す．

2. シミュレーション時に衝突を検知した場合，障害物の位置と高さが経路決定部にフィードバックされ，この情報を基に別経路が決定され，シミュレートを継続する．ここで，オペレータの適用が出来ないと判断された場合，現在の初期状態として成立している状態表記をノード作成部が，また，失敗オペレータをTMSが，成功オペレータを問題設定システムがそれぞれ受け取る．

3. ノード作成部はこれらのうち新規追加された状態表記に対し新たにノードを作成し，TMSがこれらを信念とする前提ノードをinとし，現在は成立していない状態表記の前提ノードをoutとする．

4. 次にTMSは依存関係に基づいて論理的整合性が維持されているか確認する．この探索はレベル2のネットワークに対して行われる．ここでオペレータノードの矛盾状態とは，out状態でありながら，それを支持する妥当なノードが存在しない場合である．ここで矛盾となるノードが存在する場合，システムはCONTRADICTIONを宣言し，失敗オペレータのノードがNogoodとなる．この矛盾の影響伝播はNogoodノードで止められる．

5. 問題設定システムは，4でNogoodとされたオペレータの追加リストにあたるノードをのうちで別のオペレータノードの支持ノードとなっているものを修正計画の目標状態として設定する．

6. 問題設定システムが設定した目標状態の状態表記のレベルにより，妥当な修正レベルによる計画がプランナによってなされる．プランナは目標状態達成のために妥当なレベルのオペレータ列を作成する．ここで生成されたオペレータ列がレベル1のものであれば，さらにレベル2のオペレータに展開される．

7. 5で作成された計画はその理由づけと共にノード作成部に送られ新しいノードの作成および，理由付けの追加が行われる．

8. TMSは，この理由づけに基づいて，新たなノードを追加したネットワークが論理的に無矛盾であることを調べ，なお矛盾が存在する場合5からの処理を繰り返し，無矛盾が確認された場合は修正処理を終える．

9. 最後に成功オペレータを生成したオペレータ列から除いた修正オペレータ列を移動経路計画部へ送り，経路の決定とオペレータ適用が成功オペレータの次のオペレータから進められる．

4 適用例

今，図2の問題が図6のように変化していたとする．すなわち，箱cの上に箱dが置かれた状態になっているとする．

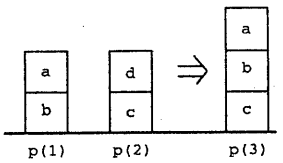

図 6: 環境変化例

移動経路計画部においては，**pickup(a,b)** と **puton(a,p(4))** の実行は成功するが，**pickup(c,p(2))** の適用については失敗を検知し(図 7)，

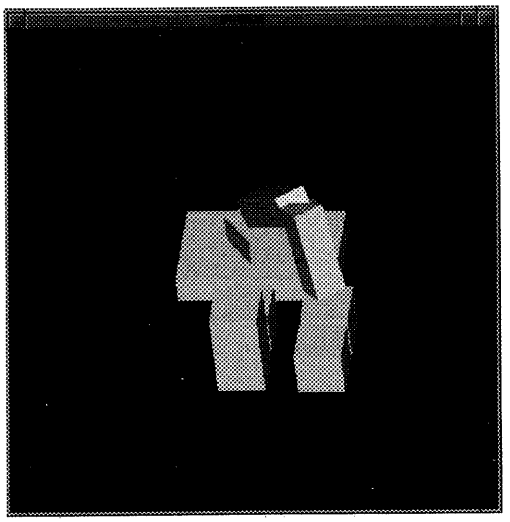

図 7: グラフィックシミュレータによる衝突検知

適用失敗オペレータ **pickup(c,p(2))** と成功オペレータ **pickup(a,b)**, **puton(a,p(4))** 及び，現在成立している状態表記がオペレータ修正部へと送られる．オペレータ修正部においては，現在

の環境と失敗オペレータ情報に基づいたノード間の依存関係の整合性より，目標状態 **hold(c)**(レベル2)が設定される．すると以下，図8 に示すように，まずレベル2オペレータ **pickup(c,p(2))** が生成される．

この場合，前提条件の **clear(c)** の未達成により，**clear(c)**(レベル1)が副目標となり，レベル1オペレータ **stack(c,p(5))** が生成される．そして，このオペレータをレベル2において展開，

pickup(d,c) - puton(d,p(5))

依存関係情報を更新後

pickup(d,c)-puton(d,p(5))-pickup(c,p(2))-puton(c,p(3))
-pickup(b,p(1))-puton(b,c)-pickup(a,p(4))-puton(a,b)

なる修正オペレータ列を移動経路決定部に伝える．なお，図8 において太線にて囲まれた部分がここでの修正範囲であり，また異階層間のオペレータ対応関係を示している．

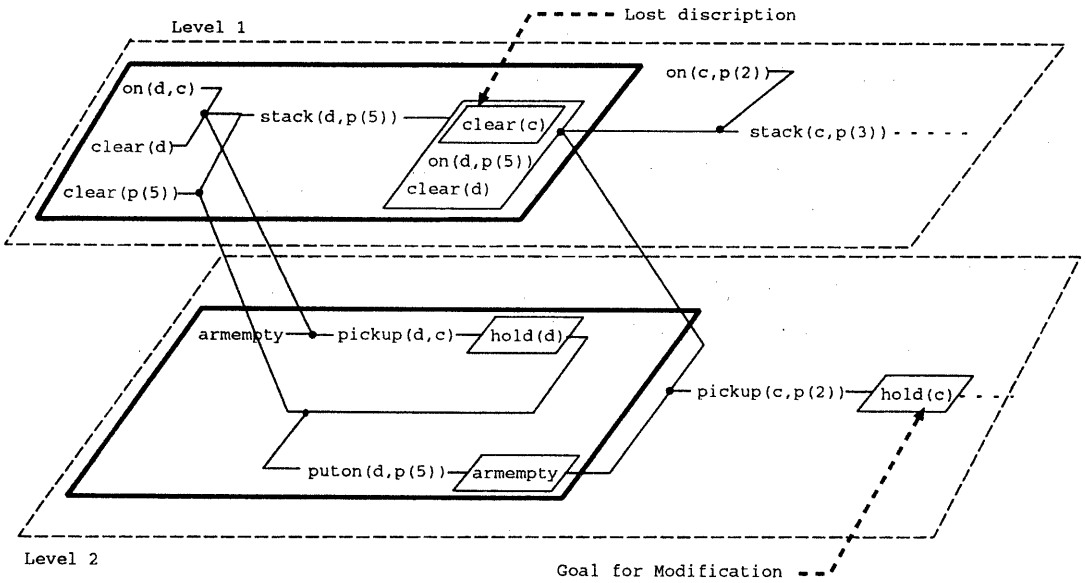

図 8: 修正例

5 むすび

本研究では，互いに直接依存し合うオペレータと状態表記間の依存関係の無矛盾性を証明系の概念に基づく局所的な無矛盾性の維持の形でＴＭＳを応用することで，計画における矛盾の影響範囲を調べることを可能とした．また，従来の修正システムの枠組みに加え，移動経路計画部の導入により，実世界にて起りうる障害の回避の手法を実現し，グラフィック・シミュレータによる衝突検知と回避機構により動的環境変化への対応を行なうことが可能となった．なお本システムはSicstus-Prolog上に構築されたＰＳＡ (Problem Solving Agent) [小川 92] 上に実現されている．

参考文献

[Doyle 79] J.Doyle, "A Truth Maintenance System", *Artifical Intelligence 12, pp.231-272(1979)*

[Fikes 71] R.E.Fikes, "STRIPS:A New Approach to the Application of Theorem Proving to Problem Solving", *Artifical Intelligence 2, pp.189-208(1971)*

[小川 92] 小川均，小島尚，今成文明，稲林昌二，大木宗一，井上浩一，
"分散型問題解決システムとその実現"
電子情報通信学会研究会資料 ,*AI92-51,pp.89-98(1992)*

[Sacerdoti 74] E.D.Sacerdoti, "Planning in a Hierarchy of Abstraction Spaces", *Artifical Intelligence 5 ,pp.115-135(1974)*

[Sacerdoti 75] E.D.Sacerdoti, "The Nonlinear Nature of Plans", *IJCAI,pp.206-214(1975)*

[Wilkins 84] D.E.Wilkins, "Domain−independent Planning: Representation and Plan Generation" *Artifical Intelligence 22, pp.269-301(1984)*

[藤田 94] 藤田智之，小川均，
"ＴＭＳの整合性管理機構を用いた積木世界のプラン修正手法"
人工知能学会誌，Vol.9,No.2,(1994)

[児玉 92] 児玉新次，
"ロボットマニピュレータのシミュレーションと３次元表示"，
立命館大学卒業研究論文（1992).