

3M-2

# プランニング・システム PLAM における サブ・プラン間相互作用の解決方法

金井 直樹

日本アイ・ピー・エム株式会社 サイエンス・インスティテュート

## 1. はじめに

近年、工程設計、ライン設計、ダイヤ作成などの非常に複雑な問題に対して、AIによるアプローチが関心を集めている。このような複雑なプランを自動的に生成するシステムをプランニング・システムとよぶ。プランニング・システムの目的は、解の探索空間を縮小すること、作成されたプラン内に存在するサブ・プラン間の競合を解決し、又、その解決方法の根拠を与えることなどがあげられる。階層的プランニング・システム PLAM (PLAN Maker) [1] は、ユーザ固有の問題に対するプランニング・システムをユーザが容易に構築できることを目的として作られた。本論文では、PLAM における、サブ・プラン間に発生した相互作用等の競合の解決方法、および、そのための知識の表現方法について述べる。

## 2. プランニング・システム PLAM

PLAM は、目標とする計画を抽象的なサブ・プランに分解し、徐々に詳細化して計画を作成する階層的なプランニング・システムであり、プランニング・システム NOAH [2] を拡張したシステムである。

ユーザは、PLAM に対して、サブ・プランに関する知識と、サブ・プラン間の相互作用などを解決するための知識である Critics を定義することができる。PLAM は、これらの知識をもとにしてプランを作成する。プランは、EP-Net (Extended Procedural Network) という、サブ・プランの半順序集合で表現される。EP-Net では、サブ・プランはノードで表される。サブ・プラン間の依存関係は、ノード間に張られた Dependency link で表される。EP-Net の例を図 1 に示す。

PLAM では、プラン作成中に問題が生じそれ以上プラン作成が不可能な場合には、バックトラックを起し他の知識を用いてプランを作成する機能を持つ。

なお、PLAM は、VM/CMS 上の Prolog (VM/Programming in Logic) で記述されている。

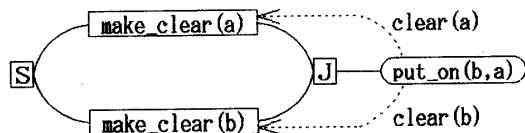


図 1. EP-Net の例

## 3. Critics

プラン作成においては、作成中に発生したサブ・プラン間相互干渉等の問題を解決する必要がある。又、より望ましいプランを作成するためには、複数のプランの中から適切なものを選択する必要がある。PLAM では、EP-Net を操作しノードの順序付けや選択などを行うことにより、これらの問題に対処する。Critics には、その Critics が対処すべき問題と、その解決法に関する知識が定義される。発生する問題は、目標とするプランの性質に依存しているため、ユーザが適切な Critics を定義する必要がある。この時、最も重要なことは、実行の結果変更されたプランの正当性が保証されるような Critics が、容易に定義できることである。変更されたプランの正当性は、サブ・プラン間の依存関係が保持されているか否かで判断できる。PLAM では、サブ・プラン間の依存関係を自動的に維持するような EP-Net 操作の述語を用意し、さらに、Critics 発動条件などの情報を簡単に定義できるようにして、ユーザが容易に Critics を定義できるような枠組を提供している。

## 4. Critics の定義

Critics は、Prolog の節の形で定義される。以下に Critics の定義を示す。

- (1) critics(Critic名, target, Arg\_pattern)  
    ← Search\_arg\_pattern.
- (2) critics(Critic名, fire, Arg\_pattern)  
    ← Execute.
- (3) critics(Critics名, when, When).
- (4) critics(Critics名, type, Type).

Search\_arg\_pattern はユーザが定義する Prolog の述語であり、その Critics が対処すべき問題を発見し、その問題に関する情報を返す。(1) が成功すると、(2) の Execute によって EP-Net を操作することにより、その問題が解決される。ただし、Execute はユーザが定義する Prolog の述語である。ユーザは、PLAM が用意している EP-Net 操作の述語を用いることにより、Execute のプログラムを記述することができる。これらの述語は、EP-Net 中の全ての Dependency link が維持されるかのチェックを行い、維持されない場合は失敗する。この結果、Critics 実行により変更されたプランの正当性が保証される。(3) と (4) には、その Critics の発動条件に関する情報が定義される。さらに、Critics には優先順位を定義することができる。以上の情報を用いて、PLAM では、(3) と (4) の条件を

Conflicts Resolution Method of Planning System PLAM

Naoki KANAI

Science Institute, IBM Japan, Ltd.

満たし、(1)が成功する最も優先順位が高いCriticsを選択して問題の解決を行い、この操作を繰り返すことにより全ての発生した問題を解決する。

5. 汎用Critics

汎用Criticsは、システム定義のCriticsであり3種類ある。resolve\_interaction\_criticは、あるノードが並列状態にあるDependency linkを破壊する可能性がある場合に、ノードの再配置を行なってDependency linkを維持する。その例を図2に示す。このCriticsは、実行の結果、Dependency linkが維持できないと失敗してバックトラックを発生させる。eliminate\_redundant\_criticとeliminate\_same\_criticは冗長なノードを除去するためのCriticsである。もし、そのCriticsの実行によってDependency linkが維持できない場合には何も行わない。これらのCriticsは、NOAHで用意されていた汎用Criticsと目的は同じだが依存関係を適切に維持する点が異なっている。

6. 問題依存型Critics

問題依存型Criticsは、ユーザが問題に応じて定義する

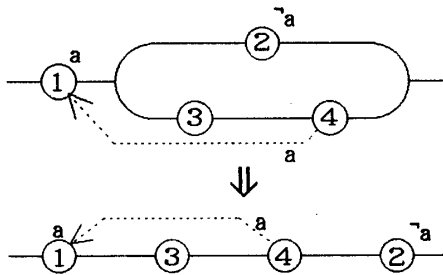


図2. Critics実行例

Criticsである。ここで例として機械部品の加工設計を考えてみる。加工設計では、面や穴を加工するために基準となる面や穴を決定しなければならない。基準となりうる部分は複数存在するので、その中から優先順位が高く、かつ、計画全体が実行可能となるものを選ぶ必要がある。これらの点を考慮して試作した加工設計プランニング・システムでは、以下の問題依存型Criticsが用意されている。

- ・面や穴の基準としての優先順位を決めるCritics
- ・なるべく多くの部分を加工できる基準を選択するCritics
- ・実行可能な順序を決定するCritics

この加工設計プランニング・システムの実行例を図3に示す。この例は5面3穴の部品の加工設計を行っている。中間状態では、非常に多くの基準の組み合わせが考えられるが、最終的には簡単なプランが作成されていることがわかる。

7. おわりに

プランニング・システムPLAMでは、サブ・プラン間の依存関係を自動的に維持するEP-Net操作述語を用意した。また、Criticsの持つ特徴を簡単に定義できるような枠組を提供した。この結果、ユーザは、問題に応じたプランニング・システムを、PLAM上に容易に実現することができる。

参考文献

[1] 金井 直樹, "依存関係を利用した階層的プランニング・システム PLAM", 情報処理学会 知識工学と人工知能研究会, 47-1, pp1-8, 1986.  
 [2] Sacerdoti, E. D., "A Structure for Plans and Behavior," Elsevier, New York. 1977.

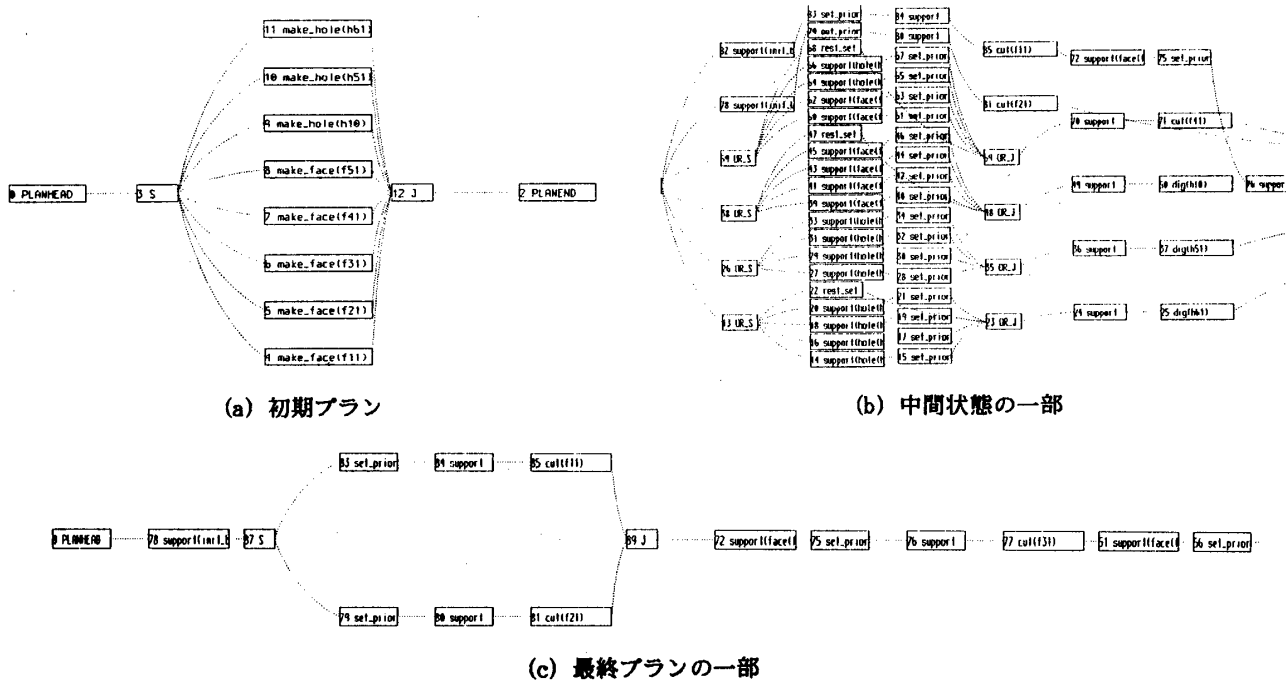


図3. 加工設計プランニング実行例