知識型計画システムにおける制約指向型説明機能†

大場 みち子†† 薦田 憲久†† 川嶋 一宏††

生産計画，作業割付等の計画問題を対象とした計画システムでは，すべての条件を取り込むことは難しいため，エンドユーザは，システムが作成した計画結果を修正することが多い．本論文では，この修正する作業を支援するため，作成された計画結果の良否を判断するための情報を出力する機能を，容易かつ柔軟に実現する汎用的な説明機能を提案する．本方式の対象となる知識型計画システムでは，対象ごとに異なる制約条件や業務特有の計算処理を簡易言語で定義し，これを利用して計画を立案している．提案方式は，この制約条件と計算処理から計画結果を説明するもので，制約指向型説明機能と名付けている．本方式では，説明のための計算処理や制約条件に応じた説明文選択の条件，説明文章等の説明内容をエンドユーザが個別に定義する．これらの定義と計画立案に用いた制約条件や業務特有の計算処理とを組み合わせて計画結果を説明する．また，説明文の選択では，制約条件間の関係等を利用して条件照合の効率化を図る．これにより，各種計画システムに対して汎用的に利用可能な説明機能を実時間で実現できる．また，従来，計画システムごとに個別のプログラムで実現していた計画結果に関する情報の取扱いを，エンドユーザでも容易に実現できることを確認した．

1. はじめに

製造，流通，公共などの多くの場で，従来，人手で実施されている生産スケジューリング，作業割当等の各種計画業務を計算機化したいというニーズが強くなっている．しかし，すべての条件を計画システムで考慮することは困難であり，時々刻々と変化する条件の入力，維持に膨大な労力を要し，その条件を加味した計画立案方法の開発も困難である．そこで，計算機を利用した計画システムでは，初期解を計算機で作成し，それをエンドユーザが人手で修正して最終的な計画とするアプローチが採られることが多い．システムが作成した計画結果をエンドユーザが修正する際に，計画結果の良否を判断する材料として，計画結果のみならず，計画結果に至った理由，考慮した制約条件，評価値等の情報が必要である．

一方，計画業務は，人間の知的な活動の1つでもあり，計画システムへの知識工学的なアプローチも多くなってきている[1)~3)]．このアプローチにおいては，最適な解を合成するための新たな推論機構のほか，知識の不足や誤りを発見するためのデバッガ的な役割を持つ機能を付加する必要がある．

このような計画結果に対するシステムからの付加的な各種の情報提供機能の代表的なものに，エキスパートシステム構築ツールに具備されている推論の過程をユーザに説明する機能がある[4)~6)]．この機能では，システムがたどった論理，すなわち，実行されたルールの連鎖を知ることができ，この説明機能により，異常原因を究明するような診断問題向けのシステムにおいては，原因の根拠を明確化できる．しかし，計画問題は，単なる3段論法的な解法によって計画を作成する問題ではないため，**図1**に示すような単にルールの連鎖を示すだけの従来の方法では，計画結果の良否や計画立案の考え方を把握することが困難である．

計画結果の良否判定に係る機能として，機械の稼働率等の各種の評価値や，個々のジョブと制約条件との関係等の情報を出力する機能等を具備した知識型の計画システムがいくつかある[7),8)]．これらの機能の実現は，必要とする情報を加工する処理や表示処理のプログラムを，時間と労力をかけて計画システムごとに個別に追加開発する必要がある．また，制約条件や計画方針等の変更時には，追加した説明のための処理も変更しなければならず，説明内容の定義・変更に対する柔軟性が問題となる．

本論文では，上記の問題を解決するため，説明内容をエンドユーザが個別に定義し，計画立案に用いた制約条件や業務特有の計算処理とを組み合わせて，制約条件との関係から計画結果を説明する制約指向型説明方式を提案し，その具体的実現方式を示す．エンドユーザが定義する説明内容は，説明のための計算処理や条件に応じて説明文を選択するための条件，説明文章等である．この説明文選択のための条件には，計画立案に用いた制約条件が流用される．

以下，2章で計画システムにおける説明機能の必要性と説明機能を実現するための課題を述べ，3章ではこの課題を解決する制約条件指向型説明機能の実現方

† Constraint-based Explanation Function for Knowledge-based Planning System by MICHIKO OBA, NORIHISA KOMODA and KAZUHIRO KAWASHIMA (Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所

ルール1：if 納期の厳しい仕事が 6 以上あり
残工程の多い仕事が 5 より多い
then 納期に着目した割付を実行
ルール2：if 納期の厳しい仕事が 5 より少なく
残工程の多い仕事が 5 より多く
割付時間の多い装置が 5 より多い
then 装置に着目した割付を実行
：

（1）ルールの例

納期の厳しい仕事が 7 で、
残工程の多い仕事が 3 で、
加工時間が長い仕事が 4 の時、
ルール1により、納期に着目した割付を実行
：
納期の厳しい仕事が 4 で、
残工程の多い仕事が 6 で、
割付時間が多い装置が 6 の時、
ルール2により、装置に着目した割付を実行
：

（2）説明例

図 1 計画システムでのルール連鎖の例
Fig. 1 An example of rule link.

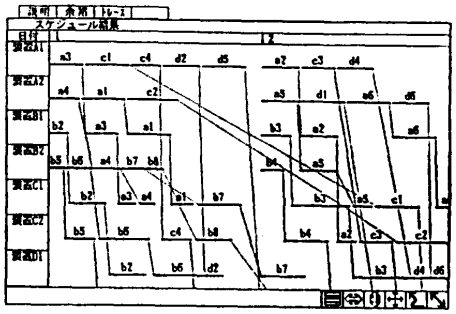
式を，4章では説明内容の定義方法を提案し，それらの定義に基づく説明処理の実現において説明文選択条件の判定処理の効率化手法を5章で述べ，6章では提案方式を実装した説明機能の計画システムへの適用例ならびにその有効性を示す。

2. 計画システムにおける説明機能

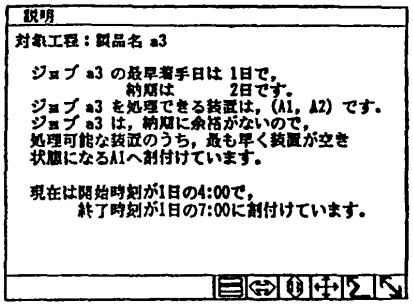
2.1 要求される説明機能

エンドユーザが計画結果を判断するためには，計画結果に関する種々の情報が必要である．スケジューリング問題の例では，①なぜそのリソースにそのジョブが割り付けられたのか，②遅らせることのできる余裕はどのくらいか，③装置の稼働率はどのくらいか，等である．これらの情報があれば，エンドユーザは，ジョブの開始時刻を遅らせたり，早めたり，代替装置への割付変更等の判断を実施できる．このような判断を適切に行うためには，単に結果を出力するだけでなく，計画結果に関する各種の説明が必要である．このため，エンドユーザ自身が必要な情報を説明文ならびに，説明文中に現れる各種定数の計算式として柔軟に定義・変更できる説明機能が必要である．

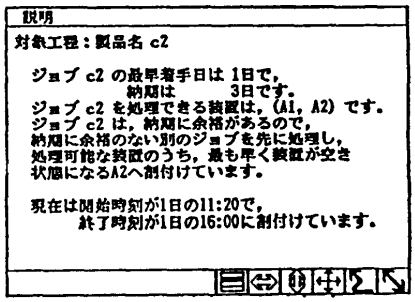
望ましい説明機能の操作イメージを**図2**を用いて示す．各ロットを1つの割付単位（ジョブ）として，時間軸上に割り付ける日程計画問題[2]では，計画結果は図2(a)のようなガントチャート形式で表示される．



（a）計画結果の表示例



（b）ジョブ「a 3」を説明対象にした説明例



（c）ジョブ「c 2」を説明対象にした説明例

図 2 説明機能の操作イメージ
Fig. 2 Operation image of the explanation function.

計画結果に対し，説明機能を起動し，ガントチャート上で説明対象のジョブとして例えば「a 3」をマウス等で指定する．説明機能は，あらかじめ入力されている説明文選択条件を判定し，成立する条件に対応する説明文名の説明文出力形式に従って説明文を生成し，別のウィンドウに図2(b)に示すような説明文を表示する．説明対象として「c 2」を指定すると，同様に図2(c)に示すような説明文が表示される．

提案方式では，エンドユーザにより，あらかじめ定義された制約条件と説明文章に基づいて，制約条件をチェックしその状況に応じて，対応する説明文を作成出力する説明機能を提供する．以下，これを制約指向型説明機能と呼ぶ．

2.2 実現のための課題

2.1 節で述べた説明機能を実現するための課題を以下に示す.

(1) 説明文章中には，必要とするデータの所在や出力形式の定義が必要である．しかし，計算機処理の専門家でないエンドユーザは，計画情報を自由に取り扱うことが困難であるため，エンドユーザが容易に理解できる計画システムとのインタフェースが必要である.

(2) 計画問題は多種多様であり，説明内容を標準化できない．計画問題ごとに制約条件や評価指標が異なる．また，それらの追加・変更も多い．また，エンドユーザの判断等により，制約条件の緩和や計画方針の変更が実施されることもあり，それぞれの場合によって支援のための説明内容が異なる．また，説明すべき内容が単なるデータではなく，業務固有の計算処理の後に得られるものが多い．したがって，エンドユーザが状況に応じて，柔軟に説明内容を定義・変更できることが重要である.

(3) 計画問題では，条件が複雑で状況に応じた説明文章を個々に定義すると，膨大な量になる．しかし，異なる状況にあっても，パラメータの値が変わるだけで，共通的に利用できる説明文もある．そこで，共通的に利用可能な説明文をいくつかの文章のまとまりとして定義でき，これらの説明文を状況に応じて切り替えることが必要である.

(4) 各種計画システムに対し，汎用的な説明機能を実現するには，計画立案部分とは独立した処理が必要となる．この場合，制約条件のチェックや計算処理等，計画立案部分で実行された処理を部分的に再現することが必要である．これらの処理は，チェックすべき制約条件も多く，計画問題特有の複雑な処理になることが多いため，効率的な処理が必要である．また，説明文作成において，これらの処理結果を取り込み，説明文章に合成するしかけも必要である.

(5) 説明文作成処理では，説明対象情報の入力処理や作成した説明文を表示するためのウィンドウの生成，説明文の表示処理等のウィンドウ処理を行うプログラムが必要である．しかし，これらの開発は，計算機処理の専門家でないエンドユーザには多大な負担となるため，開発工数の低減が必要となる.

3. 制約指向型説明機能の全体構成

知識型の計画システム[9]では，計画対象のデータを入力して，製品を出荷する納期や製品の投入順序による機能の段取替えの必要可否等の制約条件に着目して，納期の厳しさや段取時間の長さといった評価基準を定めて，それぞれの状況に応じて計画を立案する．すなわち，計画対象が着目する制約条件とその評価基準を適用して，どのような状況にあるかを判別し，その状況に応じた計画戦略（例えば，納期優先割付等）により，計画を作成する.

この知識型計画システムに制約指向型説明機能[10]を組み込んだ場合の構成を図3に示す．制約指向型説明機能では，計画立案の際に用いた制約条件や計算処理を利用し，チェックすべき制約条件と，それを満たした時に表示する説明文名を定義した説明文選択条件に基づいて，説明文名を選び，説明文名が示す説明文の出力形式に従って，説明文を作成し，表示する．説明文作成の論理に関する定義は対象の専門用語で記述することができる.

以上の説明機能の入力情報となる説明文作成の論理を，以下，説明論理と呼ぶ．なお，説明文作成処理で

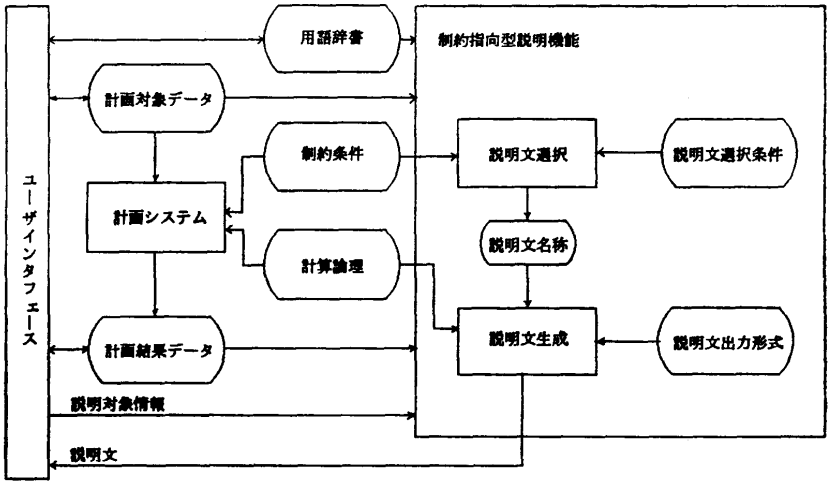


図3 制約指向型説明機能の構成
Fig. 3 Knowledge-based planning system configuration with the constraint-based explanation.

の入出力処理プログラムは，ターゲットマシン固有の入出力処理系に合わせ，標準化した部品プログラムとして用意し，利用する方式とする．

4. 説明論理の定義方法

4.1 定義の考え方

説明論理の具体的な定義に関する考え方は以下のとおりである．

（1） 計画結果の説明として単なる評価値だけではなく，文章として定義できるようにする．また，説明文章はフィルインザブランク (fill in the blank) 方式で定義し，ブランク位置にいれるべきデータは専門用語によりその値の所在あるいは計算式を定義できるようにする．

（2） その説明文章の選択は，いくつかの説明文のまとまりに名称を付け，チェックする制約条件やその評価基準を if 部に記述し，条件部を満足した時に出力する説明文章の名称を then 部に記述する．また，if 部に記述する制約条件の定義も専門用語で定義できるようにする．

（3） 計画システムごとに使用する変数に対し，専門用語を定義できる用語辞書をインタフェースとする．

4.2 定義内容

説明論理の記述内容は，(ⅰ)データ構造と変数名の定義（用語辞書）と，(ⅱ)その変数名を用いた説明論理記述言語で記述する説明文選択条件と説明文出力形式からなる説明文作成論理の定義（説明論理ソースプログラム），および(ⅲ)説明論理ソースプログラム中で使用する変数（説明変数）を算出する計算式の定義（計算論理）あるいは，(ⅳ)制約条件の定義の4種類がある．計算論理は，説明論理ソースプログラムにおける説明文章の定義で使用する説明変数を算出する計算式や対象データ等のデータの加工方法，説明文章の選択条件で使用する制約条件等の論理を業務論理記述言語[11]で定義するものである．

（1） 用語辞書

計画業務において，入出力データのデータ構造と変数名の定義は，説明論理ソースプログラムおよび計算論理ソースプログラムとは独立した表形式の用語辞書を用いる．用語辞書はエンドユーザとシステム構築者のインタフェースとなるものである．説明論理ソースプログラムおよび計算論理で使用する入出力データ名を日本語変数名と英文字変数名で定義する．

（2） 説明論理ソースプログラム

説明文作成の論理をブロックにまとめて定義する．各ブロックは，説明ブロックを定義する．

explain block 文，説明ブロック内に記述された変数が関連する対象データの識別子を定義する for 文，ブロック入力変数を定義する environment 文，説明文章の名称と出力形式を定義する expression 文，説明文章の選択方法を定義する select 文から記述される．select 文は，判定すべき制約条件の真偽の条件を記述する if 部と，if 部の条件を満足した時出力する説明文名を記述する then 部からなるルールで定義する．なお，説明論理中でのみ使用される変数の加工方法（計算式等）を計算論理と同様の logic 文で定義することも可能である．

日程計画問題[9]を例に説明論理の定義例を図4～6に示す．例えば，図4のルール1では，「最早割付条件」，「納期余裕小」および「装置割付作業多」という条件がすべて真ならば，説明文名「説明文1」という

```
explain block;
  for (x,y) where
    x is one of ロット, y is one of 装置;
  environment;
    input 装置割付負荷大(x,y), 最早割付条件(x), 納期余裕小(x)…
  end environment;
  select;
    if (最早割付条件(x) .AND. 納期余裕小(x)                    /*ルール1*/
        .AND. 装置割付作業数多(x,y))
      then display(説明1);
    if (最遅割付条件(x) .AND. (.NOT.加工時間小(x))             /*ルール2*/
        .AND.(前作業加工時間小(x).AND. 装置割付負荷大(x))
      then display(説明2);
    if (装置割付負荷大(x,y))                                   /*ルール3*/
      then display(説明3);
    if ((.NOT.最早割付条件(x))                                 /*ルール4*/
        .AND. (.NOT.(最遅割付条件(x)) .AND. 納期条件(x)
        .AND. (.NOT.(加工時間小(x)) .AND. 前作業加工時間小(x)
        .AND. 装置割付負荷大(x,y))
      then display(説明4);
    if ((.NOT.最早割付条件(x))                                 /*ルール5*/
        .AND. (.NOT.(最遅割付条件(x)) .AND. 納期条件(x)
        .AND. (.NOT.(加工時間小(x))..AND. 前作業加工時間小(x)
        .AND. 納期余裕小(x) .AND. (.NOT.後作業納期余裕小(x))
        .AND. 装置割付負荷大(x,y))
      then display(説明5);
    if (装置割付作業数多(x,y) .AND. 加工時間小(x)              /*ルール6*/
        .AND. 納期条件(x) .AND. 納期余裕なし(x)
        .AND. 装置割付負荷大(x,y))
      then display(説明6);
                          :
  end select;
end explain;
```

図 4 説明文選択条件の定義
Fig. 4 Definition of explanation sentences select rules.

```
explain block;
   for (x,y) where
      r is one of 装置, x is one of 対象ロット;
   environment;
      input 製品名(x), 装置名(y)
   end environment;
   expression(説明1);
      "納期遅れ作業減少のために、納期余裕の小さい仕事$製品名(x)%sを",
      "優先的に最早着手時刻に割り付けた。¥n";
   expression(説明2);
      "$装置名(y)%sのスループット向上のために、加工時間の小さい仕事を",
      "優先的に割り付けたため、加工時間の大きい仕事$製品名(x)%sは"
      "最遅時刻に割り付けた。¥n";
   expression(説明3);
      "$装置名(y)%sの割付負荷が大である。¥n";
   expression(説明4);
      "$装置名(y)%sのスループット向上のために、",
      "加工時間の小さい前作業の仕事を優先的に割り付け、",
      "その後に加工時間の大きい仕事$製品名(x)%sを割り付けた。¥n";
   expression(説明5);
      "納期遅れ作業減少のために、納期余裕の小さい仕事$製品名(x)%sを、",
      "納期余裕の大きい後作業の仕事より優先して割り付けた。¥n";
   expression(説明6);
      "$装置名(y)%sのスループット向上と納期遅れ作業減少のために、",
      "加工時間が小さく納期余裕のない$製品名(x)%sを優先的に割り付けた。¥n";
                              :
   end expression;
end explain;
```

図 5 説明文出力形式の定義
Fig. 5 Definition of output format for explanation.

```
block(納期余裕);
   for (x) where
        x is one of ロット;
   environment;
        input  納期(x),最早着手日(x),加工時間(x);
        output 納期余裕(x);
   end environment;
   logic;
        納期余裕(x) = 1200*納期(x) - 1200*最早着手日(x)
                    - 加工時間(x) + 1200 ;
   end logic;
          (a) 計算論理の定義

block(納期余裕小);
   for (x) where
        x is one of ロット;
   environment;
        input  納期余裕(x);
        output 納期余裕小(x);
   end environment;
   logic;
        if (納期余裕(x) .LT. 1200)
           then 納期余裕小(x) = .true.,
           else 納期余裕小(x) = .false.;
   end logic;
          (b) 制約条件の定義
```

図 6 計算論理と制約条件の定義例
Fig. 6 Definition of calculation logic and constraint conditions.

説明文を出力するというルールを表す．また，図5の説明文名「説明1」では説明対象のジョブが割り付けられた装置状況の説明に関する出力形式を定義している．図6は，納期や最早着手日といった時間的な制約条件に基づいて，納期の厳しさからロットを評価する論理である．

(3) 計算論理

計算論理は，説明論理と同様にブロック構造で定義できる．各ブロックは，ブロック名を定義する block 文，ブロック内に記述された変数が関連する対象データの識別子を定義する for 文，ブロック入出力変数を定義する environment 文，計算式を定義する logic 文から記述される．logic 文は，(a)算術演算式，(b)論理演算式，(c)関数式，(d)テーブル検索式，(e)条件選択式により，計算式を定義する．計算論理は，説明実行前にオフラインで，プリコンパイラにより計算効率の良い手続き型の実行プログラムに変更され，利用される．

5. 説明文選択条件判定方式

5.1 処理高速化の基本的考え方

計画問題においては，複数の制約条件が絡み合って解が合成される．そのため，説明文を作成する際の説明文の選択において，多数の制約条件のチェックが必要となる．また，この制約条件チェックの処理は照合処理となるため計算時間がかかる．このため，単純に条件をチェックしていけば，操作から説明文出力までの応答に時間がかかる可能性が強い．計算効率を低下させないため，次の3つの基本的な考え方により，条件照合の効率化を図る．

(1) 排他関係の利用：ある条件要素の制約条件の照合が成功したならば，異なる制約条件の照合は必ず失敗する．すなわち，ある制約条件が真ならば，ある特定の異なる制約条件は必ず偽であるような関係にある場合である．このような制約条件間の排他性をあらかじめ定義しておき，実行時に排他性を持つ制約条件の判定を優先することにより，必ず失敗する条件判定を回避できる．

(2) 包含関係の利用：ある条件要素の制約条件の照合が失敗したならば，異なる制約条件の照合は必ず失敗し，後者の制約条件の照合が成功したならば，前者の制約条件の照合は必ず成功する．すなわち，ある制約条件が偽ならば，ある特定の異なる制約条件は必ず偽であり，後者の制約条件が真ならば，前者の制約条件は必ず真であるような関係にある場合である．このような制約条件間の包含性をあらかじめ定義しておき，実行時に別の制約条件を包含する制約条件の判定

を先に実行することにより，必ず失敗する条件判定を回避できる．

（3） 出現期待順序の導入：複数の条件部における条件要素となる制約条件の照合回数をあらかじめ計算しておき，照合回数の多い制約条件から順に条件判定を実行する．これにより，失敗する条件判定の回数を低減する．

5.2 実現方式

上記の考え方を実現するため，説明文選択条件の全ルールにおける各制約条件の出現回数を基準に，排他関係や包含関係を持つ制約条件が優先的に判定されるように重み付けして，各制約条件の判定順序を決定する．

重み付けのための値を重み係数と呼び，出現回数とこの重み係数を掛けた値を各制約条件の得点とし，得点の多い制約条件から順に判定する．重み係数は，上記の考え方より他の制約条件との依存関係がない独立の制約条件の場合，基準値として“1”を与える．排他関係にある制約条件には，独立の制約条件より優先的に判定されるように，基準値より大きい値（例えば“1.5”）を与える．また，包含関係にある制約条件は，別の制約条件を包含する制約条件が，包含される制約条件より先に判定されるように，包含される制約条件には基準値より大きい値（例えば“1.5”）を与え，包含する制約条件にはさらに大きな値（例えば“3”）を与える．

この考え方を**図7**を用いて具体的に説明する．図7(a)では，図4の説明文選択条件の条件部をチェックすべき制約条件Aが真の時単にAと表し，偽の時 −Aと表すよう簡略化する．なお，各制約条件間の関係は，図7(b)のように定義されている．以上に基づいて各ルールの得点を計算した結果を図7(c)に示す．この得点から G→A→D→E→B→F→H→C→J→I の順に制約条件を判定すればよいことが分かる．この判定順序に基づいて，各ルールの判定条件を示したものを**図8**に示す．

各ルールの条件照合は，決定した判定順序に従って制約条件の真偽を判定し，各ルールの判定条件との照合をすべてのルールの成立，不成立が確定するまで行う．その際，排他関係にある一方の制約条件が真ならば，他方の制約条件は偽と分かる．また，別の制約条件を包含する関係にある制約条件が偽であれば，包含される関係にある制約条件も偽である．図8では，各ルールにおいて判定すべき制約条件が真ならば“1”，偽ならば“−1”とし，その判定条件が条件成立の際の最終判定条件ならば○で囲んである．

以上により，説明文選択の条件照合では，不要な判定処理を回避し，必要な判定のみを最小限行うことにより判定処理の効率化が図れる．

ルール1： A & B & C
ルール2： D &−E & F & G
ルール3： G
ルール4：−A &−D &−E & F& H& G
ルール5：−A &−D &−E & F& H& I & G
ルール6： C & E & H & J& G

(a) 各ルール条件部の簡易表現

関係1： B ⊃ J
関係2： A⇔¬D & ¬A⇔D

(b) 制約条件間の関係

ルール / 制約条件	1	2	3	4	5	6	出現回数	重み係数	得点合計	判定順位
A	●			●	●		3	1.5	4.5	2
B	◎						1	3	3	5
C	□					□	2	1	2	8
D		●		●	●		3	1.5	4.5	3
E		□		□	□	□	4	1	4	4
F		□		□	□		3	1	3	6
G		□	□	□	□	□	5	1	5	1
H				□	□	□	3	1	3	7
I					□		1	1	1	10
J						○	1	1.5	1.5	9

●:排他関係 ◎○:包含関係 □:その他(独立)

(c) 制約条件判定順位決定表

図7 ルール条件部の制約条件判定順位の決定方法
Fig. 7 Sorting method for checking conditions.

判定順位	判定制約条件	ルール条件部 1	2	3	4	5	6	関係	付加処理
1	G		1	①	1	1	1		
2	A	1			−1	−1		●→D	Aの時ルール2不成立
3	D		1		−1	−1		●→A	
4	E		−1		−1	−1	1		
5	B	1						◎→J	−Bの時ルール6不成立
6	F		①		1		1		
7	H				①	1	1		
8	C	①							
9	J						①	○→B	
10	I					①			

1:真 −1:偽 ○:最終判定条件 ●:排他関係 ◎○:包含関係

図8 制約条件の判定順序とルールの判定
Fig. 8 Checking order and rule testing result.

5.3 動 作 例

前述の図7，8の例を用いて提案手法の動作例を次に示す．

今，設定された説明対象に対して，制約条件A，C，E，G，H が真の場合を考える．

まず，判定順位1番目の制約条件Gから始める．Gは真なのでルール2～6の照合が成功し，ルール3は最終判定条件が成立したので，条件が成立したとして判定対象から外す．2番目の判定では，制約条件Aは真なのでルール1の照合が成功し，ルール4，5の照合が失敗するので判定対象から外す．また，制約条件Aの排他関係にある制約条件Dはこの時点で偽となることが決定するので，制約条件Dが真という条件を含むルール2を照合失敗として判定対象から外す．3番目の制約条件Dの照合は判定対象のルールがないので次の判定に移る．4番目の制約条件Eの判定は真なのでルール6の照合が成功する．5番目の制約条件Bの判定は偽なのでルール1の照合が失敗し，判定対象から外す．また，包含関係にある制約条件Jもこの時点で偽ということが判明するので，制約条件Jが真という条件を含むルール6を照合失敗として判定対象から外す．ここで，判定対象がなくなるので照合処理を終了する．

本例の場合，提案方式では制約条件の照合処理は，5回である．これに対し，ルールに記述された順に制約条件を判定し，判定した制約条件に対する照合が失敗したルールを不成立として判定対象から外して，判定対象がなくなるまで照合を繰り返す方式(単純方式)では，制約条件の照合処理は10回である．

なお，図7(a)のルールを用いて，全制約条件の真偽の全組合せ 1024 ケースを生成して，提案方式と単純方式との条件照合回数を比較した結果を**表1**に示す．この結果，総計の照合回数は，単純方式が 8704 回で，提案方式が 4784 回となり，提案方式では単純方式の 55% の照合回数となり照合処理の効率化が図れる．

表1 条件照合回数の削減
Table 1 Reduction of checking operation.

制約条件数	10
実行ケース数	$2^{10}=1024$
単純方式照合回数 (A)	8704
提案方式照合回数 (B)	4784
増減率 (%) (B/A * 100)	55

表2 説明機能の適用評価
Table 2 Evaluations through applying planning systems.

計画問題	応答時間 (msec)	記述量 (行)	
		説明論理	手続き型言語
日程計画	184	343	1906
食品製造計画	223	494	2484
配合槽スケジュール	148	235	1155
平均	185	357	1848

6. インプリメンテーション

本提案の制約指向型説明機能を組み込んだ計画システムを 32 bit ワークステーション上にインプリメンテーションした．3種類の計画システム[9),12)]に適用した結果を**表2**に示し，以下の評価を得た．

説明機能の処理において，平均応答時間は数十 msec であり，エンドユーザが待ちを感じることもなく，特に問題はない．なお，論理解析処理およびプリコンパイル処理に数秒を要するが，オフラインで実行されるので，使用時には問題とならない．

記述能力については，代表的な計画問題であるジョブ・ショップ型および連続量を扱う問題に適用し，必要とする説明内容は概ね分かりやすく定義できた．各問題での説明論理は，手続き型言語で記述した場合に比べて，平均 1/6 の記述量で定義できた．説明論理を作成するための開発工数は，対象の計画問題を理解していれば，用語辞書作成を含め，手続き型言語の知識がないエンドユーザでも，1～2日で作成可能である．また，説明論理を変更した場合，本提案方式では説明論理ソースプログラムを変更するだけであり，数分のオペレーションで対応できる．これに対し，手続き型言語で実現した場合にはプログラムの変更だけでなく，コンパイル・リンクが必要となる．さらに，手続き型の言語の場合には可読性が低いため，デバック時間も必要である．手続き型言語で同様の説明機能を実現しようとした場合，計画システムのプログラムで使用している変数のデータ構造を理解する必要があるほか，使用する手続き型言語の知識も必要となり，開発工数は本提案方式の数倍になると考えられる．

7. むすび

生産計画，作業割付等の計画システムにおいて，計画結果を説明する汎用的な説明機能を提案した．本提

案方式では説明文の出力形式や状況に応じた説明文の選択方法，制約条件等を，別に定義した専門用語を用いて非手続き型の言語で記述し，これに基づいて計画結果に至った理由や計画作成時に考慮した制約条件等を説明する．これにより，各種計画システムに対して汎用的に利用可能な説明機能を実現できる．また，従来，計画システムごとに個別のプログラムで実現していた計画結果に関する情報の取扱いを，エンドユーザでも容易に実現できることを確認した．

謝辞 本研究の機会を与えて頂いた(株)日立製作所システム開発研究所 堂免信義所長，中尾和夫博士に感謝いたします．また，説明機能の開発にあたり，ご指導，ご協力を頂いた同・情報システム工場の方々に感謝します．

参考文献

1) 特別レポート 着実に実用化が進む計画型エキスパートシステム，日経 AI, 1988.1.18 号付録 (1988).
2) Bruno, G. et al.: A Rule-based System to Schedule Production, *IEEE Comput.*, Vol. 19, No. 6, pp. 32-39 (July 1986).
3) Arieh, D. B. et al.: Knowledge Based Routing and Sequencing for Discrete Part Production, *Journal of Manufacturing Systems*, Vol. 6, No. 4, pp. 287-297 (1987).
4) Hayes-Roth, F. et al.: *Building Expert System*, Addison-Wesley (1983).
5) 田口ほか：エキスパートシステム構築ツール XPT (2)—推論方式—，第35回情報処理学会全国大会論文集，pp. 1705-1706 (1987.3).
6) 宮元ほか：エキスパートシステム開発ツール MASTER (4)—診断問題向け機能—，第39回情報処理学会全国大会論文集，pp. 191-192 (1989.10).
7) 沼尾ほか：スケジュール作成支援エディタ，第35回情報処理学会全国大会論文集，pp. 1647-1648 (1987.3).
8) 吉井ほか：FMS スケジューリングエキスパートシステム，1989年度人工知能学会全国大会 (第3回)，pp. 605-608 (1989.10).
9) 原ほか：知識型スケジューリングシステムの開発，日立マイコン技報，Vol. 4, No. 1, pp. 56-61 (1990.3).
10) 大場ほか：知識型計画システムにおける計画結果説明機能，第40回情報処理学会全国大会論文集，pp. 240-241 (1990.3).
11) 川嶋ほか：知識型計画支援システム向業務論理記述言語用プリコンパイラ，情報処理学会論文誌，Vol. 28, No. 9, pp. 975-986 (1987).
12) 原ほか：食品製造工程向知識型スケジューリングシステムの開発，SICE 第4回システム工学部会研究会資料，pp. 1-6 (1989.11).

(平成2年3月26日受付)
(平成2年9月11日採録)

大場みち子 (正会員)

昭和57年日本女子大学家政学部家政理学科（物理学専攻）卒業．同年，(株)日立製作所入社．システム開発研究所にて，製造向計画支援システム，ソフトウェア開発プロジェクトの計画・管理システム等に関する研究に従事．現在，ソフトウェア工場にて，統合 OA システム，グループウェアの研究・開発に従事．計測自動制御学会会員．

藤田 憲久 (正会員)

昭和25年生．昭和47年大阪大学工学部電気工学科卒業．49年同大学院修士課程修了．同年(株)日立製作所に入社．システム開発研究所にてシステム計画技法，システム構造化技法，ペトリネットなどの事象駆動型システム，生産・流通業向情報処理システムなどに関する研究に従事．現在同研究所第1部主任研究員．56～57年 UCLA に留学．工学博士．IEEE，電気学会，電子情報通信学会，計測自動制御学会などの会員．61年度計測自動制御学会論文賞，62年度計測自動制御学会技術賞受賞．

川嶋 一宏 (正会員)

昭和35年生．昭和57年東京理科大学理工学部機械工学科卒業．昭和59年同大学院修士課程修了．同年(株)日立製作所入社．現在同社システム開発研究所にて，離散型システムの計画・制御技法の研究に従事し，知識型計画支援システムの開発を担当．日本機械学会，計測自動制御学会各会員．