

2D-7 知識型計画システムにおける
計画結果説明機能

大場 みち子,* 薦田 憲久,* 川嶋 一宏*
* (株)日立製作所 システム開発研究所

1. まえがき

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、実際の計画問題における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。計画結果を修正する際には、計画結果の良否を判断する材料として、システムが作成した計画結果の根拠等の説明が必要である。これに対し、一般のエキスパートシステム構築ツールでは、実行されたルールの連鎖を示して、推論の過程をユーザに理解させる説明機能が具備されている。しかし、計画問題は、単なる3段階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、ルールの処理からは実現できず、計画システムごとに個別にプログラムを作成しなければならないという問題があった。これに対し、計画立案に用いた制約条件の定義を用いて、計画結果を説明する汎用的な説明機能を開発したので報告する。

2. 計画システムにおいて要求される説明機能

エンドユーザが計画結果を修正する場合、その計画結果を判断するためには、計画結果だけではなく、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の情報が必要である。スケジューリング問題では、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の情報に関する説明があれば、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適切に実施できる。しかし、計算機の専門家でないエンドユーザが、計画情報を自由に取扱うことは困難である。また、要求される説明内容は、対象とする計画問題や計画方針等の変更に応じて変化する。そのため、それぞれの状況に応じて、エンドユーザでも、柔軟に説明内容を定義、変更できる説明機能が必要となる。

3. 説明機能の基本的な考え方

知識型計画支援システム (HPGS:Hitachi Flexible & Intelligent Planning Support System)¹⁾ を利用した知識型スケジューリングシステム²⁾ では、計画条件 (制約条件や目的関数等) を「用語辞書」(図1) に定義した専門用語を用いて、簡易言語 (業務論理記述言語) で記述する (図2)。計画の立案は、この計画条件を満足する解を、「計画の状況を判断し、その状況に応じた解法による割付を実行する」という手順で求める。簡易言語で記述された計画条件は、プリコンパイラで手続き型の実行プログラムに変換される。

本提案方式では、対象業務ごとに異なる制約条件や、業務特有の計算式の定義 (以下、計画論理と呼ぶ) は、計画立案時に定義したものを利用する、一方、説明文については、説明内容の定義 (以下、説明論理と呼ぶ) を簡易言語 (説明論理記述言語) で記述する (図3)。説明論理では、関連する対象データの識別子を定義するfor文、入力変数を定義するenvironment文、説明変数を定義するdefinition文、説明文章の名称と出力形式 (文章) を定義するexpression文、説明文章の選択方法を定義するselect文から構成する。select文の選択条件には、専門用語または計画論理で定義した制約

日本語変数名	英文字変数名	属性	領域
ロットNO.	LOT_NO	数値	4
製品名	PNAME	文字	10
納期	NOUKI	数値	4
最早着手日	CHAKU	数値	4
:	:	:	:

図1 用語辞書

```

block(納期の厳しい製品);
for (x) where
  x is one of ロット;
environment;
  input 納期(x),最早着手日(x),...
  output 納期の厳しい製品(x);
logic;
  if (納期余裕(x).LT.600) /* 分 */
  then (納期の厳しい製品(x)=true;
  納期余裕(x)=納期(x)-最早着手日(x)
  -加工時間(x);
  :
  
```

図2 計画論理の記述例

条件等を記述できる。なお、説明文中でのみ使用される変数の計算方法を、計画論理と同様のlogic文で定義することもできる。提案方式では、これらの定義を用いて指定された条件に対応する説明文を作成し、出力する。なお、説明論理で使用する専門用語は計画立案時に定義した用語辞書を用いる。

4. 説明機能の全体構成

提案する説明機能の全体構成を図4に示し、その処理概要を以下に説明する。なお、論理解析部およびプリコンパイラの処理は、オフラインで実行し、計画立案時には、説明文生成部の処理のみを実行する。

(1) 論理解析部 説明論理と計画論理の字句解析を行い、構文をチェックする。構文エラーがなければ説明文選択条件、説明文の出力形式等をテーブルに登録する。

(2) プリコンパイラ 計画論理と説明論理のlogic文および用語辞書を入力して、手続き型プログラムを生成する。

(3) 説明文生成部 計画システムでの計画立案時に、説明機能が起動されると、すべての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したがって説明文を生成し、ディスプレイに出力する。説明文出力形式に説明変数が定義されていれば、エンドユーザが計画結果の表示画面から直接指定した説明対象の情報と用語辞書に基づいて対応する英文字変数名の値を出力する。また、説明変数が計画論理または説明論理中のlogic文で定義されていれば、プリコンパイラが生成した手続き型プログラムを実行し、その結果の値を出力する。

5. あとがき 本方式により、各種計画システムに対して汎用的な説明機能を実現でき、従来、計画システム毎に個別のプログラムで実現していた計画結果に係る情報の取扱いを、エンドユーザにも可能とした。

参考文献 1) 川嶋,他: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,情報処理学会論文集, vol.28, No.9, pp975-986 (1987,9), 2) 薦田,他:知識型計画支援システムHPGSによるスケジューリングシステム,第15回システムソフトウェア・第10回知識工学ソフトウェア合同ソフトウェア講演論文集, pp361-366 (1989,10)

```

explain block;
for (x) where
  x is one of ロット;
environment;
  input ロットNO(x),製品名(x),...
end environment;
select;
  if (ロットNO(x).NE.0)
  then display(説明文1);
  if (納期が厳しい製品(x).EQ.true)
  then display(説明文2);
  if (納期が厳しい製品(x).EQ.false)
  then display(説明文3);
  :
definition;
  $job = 製品名(x);
  $seq = 対象工程#;
  $est = 最早着手時刻(x);
  :
expression(説明文1);
  "対象工程:製品名$job%$sの第$seq工程$fn",
  "ジョブ$job%$sの最早着手日は$est%2dで,$fn",
  " 納期は$est%2dです。$fn",
expression(説明文2);
  "ジョブ$job%$sを処理できる装置は,",
  "($mac1%$s,$mac2%2d)です。$fn",
  "ジョブ$job%$sは納期に余裕がないので,$fn",
  "処理可能な装置のうち,最も早く装置が",
  "空き状態になる$amacへ割付ています。",
  :

```

図3 説明論理の記述例

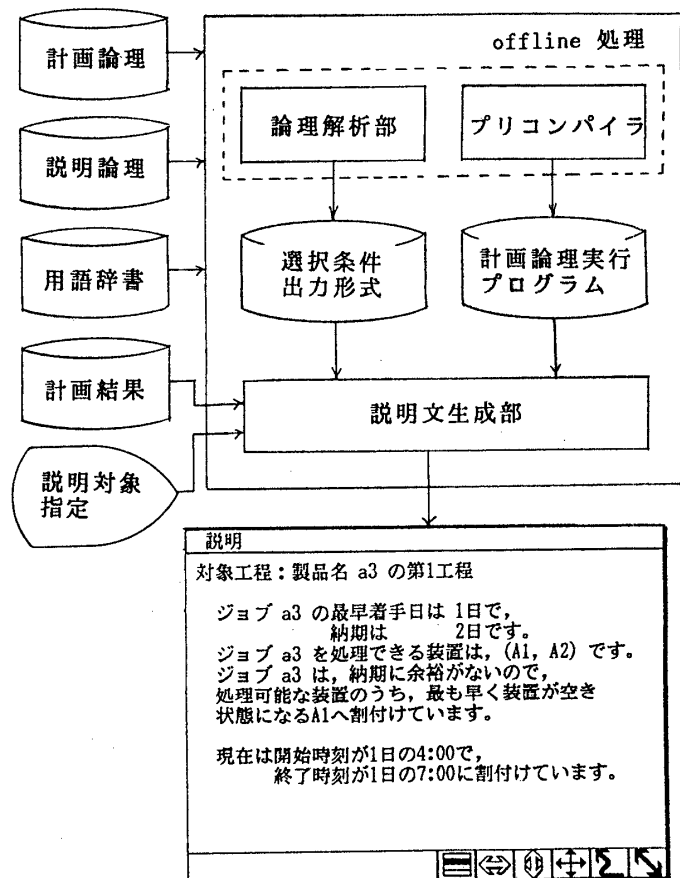


図4 説明機能の全体構成