

## 2L-6

E S H E L L を用いた  
スケジューリングシステムの評価

稲田 裕 \* 菊地 智 \* 平岡 薫 \* 飯島 信夫 \* 山口 隆義\*\*  
\* 富士通 株式会社 \*\* ㈱ 富士通愛知エンジニアリング

## 1. はじめに

我々は、富士通沼津工場ソフト開発センターの計算機オープン利用時間割当てを行うエキスパートシステム：CASS / X (Computer Assisted Scheduling System/ eXtended) を、E S H E L L を用いて開発している。

本システムの開発契機及びプロトタイプの開発経緯については、第32回情報処理学会全国大会で既に報告した。今回は、プロトタイプから実用システムへの拡張経緯、拡張されたシステムの評価、及び実用化に向けての問題点とその対処法について報告する。

## 2. システムの拡張

本システムでは、まずプロトタイプとして、M380-7号機上の仮想計算機13台(M380#7AVM)に関する割当てのみを行うシステムを開発し、十分なテストを行った上で、さらに知識を追加して全ての計算機の割当てを行う実用システムに拡張するという方法を採用した。(図1)

この拡張作業は次の2つの段階に分かれる。

まず、計算機の割当て順序に関する知識を追加し、全ての計算機に対してM380#7AVMと同じ方法で割当てを行うシステムに拡張した。これに伴う変更は、修正前のKS (Knowledge Source : 知識源) 28個、ルール総数241に対し、2つのKSの追加(ルール総数36)と、スケジューリングデータをメモリ上にロードする関数の修正のみに止まった。

次に、割当て範囲を拡張したことによる知識の追加・修正を行った。この作業では、特殊な計算機に対する処理を追加する他に、割当て方法自体にも多くの修正が必要となった。これは、M380#7AVM に対しては暗黙の前提となっていた条件が、他の計算機に対しては成立しない場合がしばしばあったためである。例えば、M380#7AVM の場合には、稼働時間と申請時間がほぼ等しいという性質を利用して割当て時間を決定できたが、他の計算機ではこのような前提は使えない。

この作業に伴い、殆ど全てのルールが修正を受け、KSの構成も若干変更された。この結果、現在のKS数は35、ルール総数294となっている。

ただし、このような大規模な修正ではあったが、知識がKSという形にモジュール化されていたため作業は比較的容易

であった。また、拡張作業を通じて黑板(データ構造)はほとんど変更を受けなかったことは注目に値する。

## 3. 評価

CASS / X の動作を分析したところ、熟練者(エキスパート)と比較して次のような点で同等または優れていることが分かった。

## 1) 処理時間

熟練者が手作業で行えば3日かかる作業を、CPU時間5分前後で処理する。

## 2) ケアレスミス

熟練者による作業では、申請の見落としや指定の見落とし等のため、割当て作業を最初からやりなおしたり、大幅な修正が必要となる場合が発生する。CASS / X ではそのようなことは起こらないであろう。

## 3) 空き時間が多い場合の割当て

一般的な運用形態を持つ計算機で、空き時間が十分多い場合に、セッション、月の前半/後半、昼の時間帯と夜の時間帯のバランスを考慮しながら、しかも後の申請の割当てがうまく進むように、割当てを行うことができる。一方、熟練者と比較すると次のような点が劣っている。

## 1) 特異な運用形態を持つ計算機の時間割当て

例えば、運用時間が深夜時間帯に集中しているような計算機に対して、昼と夜の割当て時間の配分を誤ったり、稼働時間に比べて申請時間が非常に少ないような計算機に対して、割り当てる時間帯の判断を誤ったりする。

これは、現在の知識がM380#7AVMの運用形態を念頭に置いたものになっているためであると思われる。

## 2) 改善の割当て

何か制約を無視しなければどうしても割当てを行うことができないような場合、CASS / X は、制約条件を大きく逸脱してしまう場合がある。

これは、制約が達成できないような場合に働く知識が少ないことが最大の理由である。さらに、CASS / X は割当て結果を評価するためのパターン認識機構を持っていないために、大局的な視点で結果を評価することが困難であることも原因の一つと考えられる。

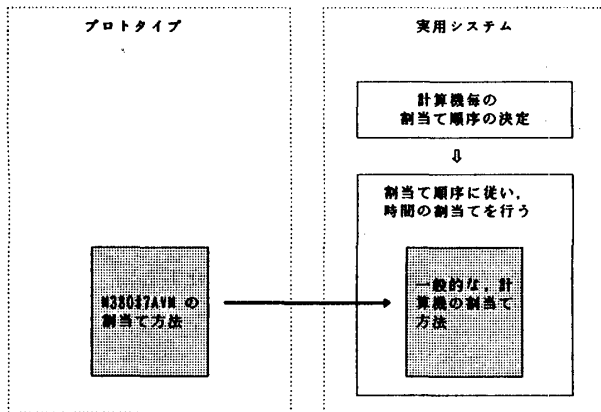


図 1

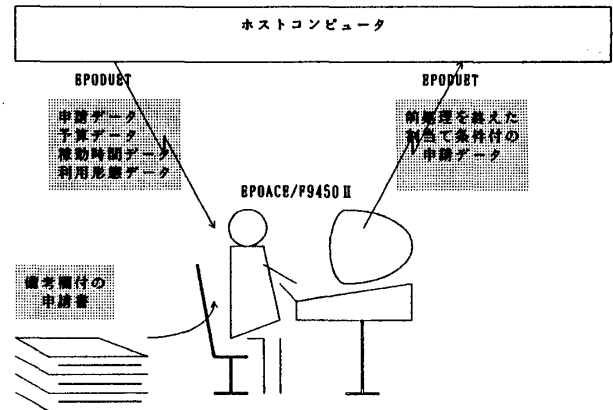


図 2

#### 4. 環境整備

##### 1) 入力データの作成支援

人間が時間割当て作業を行う場合、その入力には多種の情報から成り、しかもその表現法としては多くの形式が容認される。例えば、あるAVMを指定する方法でも、その指定を日本語で書いたもの、必要とする資源を指定することにより間接的にあるAVMを指定するもの、端末の場所で指定するもの等様々である。

このような入力を全てエキスパートシステムが受け付けるようにすることは、次の2点から非常に困難であった。第1には、多くの情報が日本語で書かれておりそれを解析することが困難であること。第2には、このような多くのデータから適当な情報を導出するためには膨大な知識が必要になるということである。

そこで、図2に示すようなCASS/X入力データ作成システムを、簡易事務処理ソフトウェアEPOACEを用いて開発した。このシステムは、熟練者を支援して、申請書に記入された指定をCASS/Xが解釈可能な入力形式に変換する為のものである。

このようにして、割当て作業は機械が行い、申請の解釈という高度な知識を要する部分は人間が行うという、分業体制が整った。

また、このシステムは、総申請時間数を各計算機の稼働時間内に収める「前処理」作業の支援も行い、これにより時間割当て作業の殆ど全ての過程が計算機によってサポートされるようになった。

##### 2) 保守ツールの開発

計算機の追加、削除、割当て順序の変更等は、非常に頻繁に発生する。そこで、CASS/Xの内部を知らなくてもこのような変更に伴う修正が容易にできるよう、対話型の保守プログラムを開発した。

また同様に、割当て順序の決定基準の変更に伴う修正を支援するようなツールも開発した。

#### 5. 実用化作業に関する考察

##### 1) プロトタイプの拡張

プロトタイプに実現されている知識のほとんどは、そのままの形では実用システムに利用することはできず、多くの修正を行う必要がある。これは、プロトタイプを開発するために設定された問題範囲が、それに固有の偏向を持っているためである。

但し、問題範囲の設定を注意して行えば、データ構造に対する修正はかなり抑えることができる。また、プロトタイプの知識表現が十分にモジュール化されていれば、この作業は比較的容易になる。

##### 2) 入力データについて

人間は多様な形式の入力を受け付けることができる。これに対し、エキスパートシステムに入力できる情報は限られているため、エキスパートシステムが専門家の作業を代替するにあたっては、入力の柔軟性を失わないように何らかの対策が必要になると思われる。

##### 3) 保守の支援

システムの運用担当者は、一般に、システムの内部に直接手を入れることに大きな抵抗を持つ傾向がある。このような状況では、AIツールの持つ編集機構の他に何らかの保守ツールが必要となる。AIツールは、このような保守ツールを開発する際の手掛かりを提供すべきである。

ESHELLの場合、知識の内部形式が公開されており、それがLISPを使って容易に操作できるリスト形式であったことは、今回の保守ツールの開発を非常に容易なものにした。

#### 6. おわりに

エキスパートシステムの、プロトタイプから実用システムへの拡張経験と拡張されたシステムの評価、及び若干の考察を報告した。今後は、従来システムとの並行運用を通じて更に改良を加え、実用化を目指す予定である。