

計算機室レイアウト用エキスパート・システムの開発†

渡辺 俊典^{††} 佐々木 浩二^{††} 安信 千津子^{††}
 永井 義明^{††} 飯塚 由美子^{††} 山中 止志郎^{†††}
 山越 実^{†††} 田中 隆^{†††}

電子計算機システムの技術サービスに従事するシステムエンジニアの業務の一部に、計算機システムの計算機室内への配置計画作業がある。この作業を知識工学的手法によって自動化するエキスパート・システムを開発し実用化した。システムは、配置物形状、配置場所選定用ルール、配置上の制約条件、各種の幾何算法を実行するプログラム等から成る知識ベースと、知識ベース内の情報を解釈・実行するための推論機構および、配置計画状態等を記憶するワーキング・メモリから成る。配置物の名称と部屋形状その他若干の情報を与えることにより、自動的に、各種の制約条件を守る配置計画を作成させることができる。本システムは UTILISP 上にて実現している。知識ベース内のルールを人手によって作り込むことの不便さを解消するために、一部のルールはルール生成用エディタによって効率良く作成できるようになっている。

1. ま え が き

電子計算機システムの技術サービスに従事するシステム・エンジニアの業務の一部に、計算機室への計算機システムの配置計画を立案する作業がある。作業の前提として、部屋の形状と配置するシステム機器の名称とが与えられ、これらの配置を、配置物相互の位置に関する制約を満足するように配置し、製図したのち、顧客に提示する。

この作業を実施するために、配置物の形状平面図や配置時の制約条件を定義したマニュアルが用意されており、システム・エンジニアはこのマニュアルを用いて机上で型紙を試行錯誤的に配置し、良好と思われる配置計画を得た段階で、製図用 CAD プログラムを用いて配置計画図を清書する。

多忙なシステム・エンジニアにとって、マニュアル上の注意事項をすべて記憶して、妥当な配置計画を作成することは骨の折れることであり、その計算機処理化が望まれる。しかしながら、計算機室の形状や計算機システムを構成する機器は、顧客によってまちまちであり、また、配置物の位置は、部屋の形状、出入口等の属性、既配置物の配置状況等に依存して決定されるため、決定的手続きによってプログラム化すること

はむづかしい。

配置問題は従来より、オペレーションズ・リサーチ分野の研究者らによって研究されてきた。数値計画法を用いたスペース・プランニング問題の解法などはその典型的な例であり、配置可能場所に物体を配置したときの配置の良否を評価する関数を最適化する配置案を求める。

この方法の良い点は、配置計画をグローバルな観点から最適化できるという点であるが、他方で配置物相互の間に課せられた制約等が数理的定式化にのりにくい場合には、問題のモデル化自体が困難となる。

これに対して、業務専門家の経験的知識を計算機上に移植するとともに、記号処理技術を活用する知識工学的接近法を用いることにより、従来は定式化することさえ困難であった問題を計算機処理化することが可能となってくる。

2. 構 想

2.1 問題の説明

計算機室への機器配置問題を下記に説明する。

(1) 前提条件

配置計画の立案にあたって、部屋形状平面図、部屋の出入口、配置にあたって障害物となる柱の位置と形状、フリーアクセス（床面タイル）の状況、他機器の基準となる中央処理装置（CPU）とコンソールディスプレイ装置（CD）の配置および配置すべき機器の名称が与えられる。

(2) 配置制約条件

配置物を部屋内に配置するに際して下記の条件を満

† Design of an Expert System for Computer Room Layout by TOSHINORI WATANABE, KOJI SASAKI, CHIZUKO YASUNOBU, YOSHIKI NAGAI, YUMIKO IZUKA (Systems Development Laboratory of Hitachi), TOSHIRO YAMANAKA, MINORU YAMAKOSHI and TAKASHI TANAKA (Software Works of Hitachi).

†† (株) 日立製作所システム開発研究所

††† (株) 日立製作所ソフトウェア工場

足させる必要がある。

- 機器は室内に配置する。
- 機器同士は配置平面図が重複してはならない。
- 機器の保守エリアを確保する必要がある。
- フリーアクセスに、機器のケーブル穴を揃え、ケーブル配線時に不都合が生じないようにする必要がある。
- 機器の前面線を揃えて配置する必要がある。
- 類似のタイプの機器は近くに配置する必要がある。
- 低速入出力装置（カードリーダーなど）は、コンソールディスプレイ装置の近くに配置し、計算機オペレータの機器操作作業の効率を向上させる。
- ラインプリンタ装置は出入口の近くに配置し、外部ユーザが利用しやすいようにする。

(3) 結 果

上記(1)で与えた部屋の内部に、指定された配置物体を、上記(2)の条件を満足するように配置し、結果の平面図を出力する。

2.2 問題解決方式の検討

(1) 数理計画法によるアプローチ

上述した問題に類似のものは、従来オペレーションズ・リサーチ分野で研究されてきた。有限面積の鉄板から切りくずの量をできるだけ少なくするように形状物を打ち抜く板取り問題、有限平面内に各種の物体を配置するスペース・プランニング問題などが、その例である。スペース・プランニング問題を例にとってその特徴を考察する¹⁾。

● 問題のモデル化

スペース・プランニング問題は、オペレーションズ・リサーチの分野では次のように定式化される。

$$\text{Min} \sum_{\rho \in S} \sum_i f_{i\rho(i)} + \sum_i \sum_j q_{ij} C_{\rho(i)\rho(j)}$$

ここに、 i, j は配置物の番号、 $\rho(i)$ は i を配置する場所の番号、 $f_{i\rho(i)}$ は i を $\rho(i)$ に配置するときのコスト、 q_{ij} は i と j が 1 単位距離離れているときのコスト、 $C_{\rho(i)\rho(j)}$ は i, j の配置位置間の距離、 S は配置物を配置位置に対応づけるすべての組合せの集合である。

● 解法

配置物の一つピックアップして、上記の目的関数値の増分の最も少ない場所に配置するという操作を繰り返して一応の解を得たあと、任意の二つの物体を相互に入れかえるなどの方法で、上記の目的関数の値を改

善することのできる解を見いだす。

● 考察

本方式は、上述した目的関数を最小化する範囲では妥当な解を導けるが、2.1 節(2)で述べた各種の条件を上記の目的関数に表現することは大変むづかしい。逆に言えば、現実の問題のもつ諸側面の内の数理的定式化の許される側面しか考慮できないという難点がある。

(2) 人工知能的アプローチ

本問題に類似の問題に対する過去の研究を考察する。

Pfefferkorn による DPS システムは、LISP 言語による家具配置問題を扱った初期のであり、形状物の LISP 言語による表現法、バックトラッキングメカニズムを用いた配置位置の探索法等を提案している²⁾。

McDermott による R1 システムは、計算機箱体への部分の組み込みを自動計画するシステムである。部品組み込み手順をルール化した知識ベースと前向き推論機能とから成る知識ベースシステムとなっているが、配置が不可能の場合には配置スペースを拡大するなど的手段によってバックトラッキングをさせている³⁾。

Roth による Errand Planner は、種々の制約のついた巡回セールスマン問題を解くシステムである。知識ベース内に経路選択用の知識等をルールの形で保有し、階層化された共通メモリ上に計画状況や、制約条件の満足状況等を記述し、非決定的にルールを呼び出して問題解決をすすめるものである⁴⁾。

以上の諸システムは、先に述べた数理計画的アプローチに比較して、問題をよりの確に表現できる点、専門家のヒューリスティクスを利用しやすく、したがって質の高い解を生成できる可能性のある点などの面で優れている。ただし、DPS は配置位置決定時に考慮する制約が限られており実用的なものではない。R1 は配置場所が確保できない場合に既配置物をずらせる等の機能がなく、本問題への適用はむづかしい。Errand Planner は問題の扱い方が参考になるが、対象とする問題が異なっている。

(3) 専門家の問題解決プロセス

システムの開発にあたって、計算室への機器配置業務専門家の作業ノウハウを調査した。その結果、機器の配置順序に関するノウハウ、配置にあたって既配置物に対する未配置物の位置を選択するためのノウハウなどが利用されており、全体としては図1の処理流れ図によって計算機処理化することができると判断し

た。図中で、配置ノウハウを用いて配置位置案を作成する段階は、決定的手続きによるプログラミングがとくに困難である。そこで、知識ベースシステムにおけるルールによって、これらのノウハウを非決定的にモデル化し、推論機能を用いてこれを活用するという方式をとることにし、前述の Errand Planner における問題解決方式を参考にしたアプローチをおこなうことにした。

3. システムの設計

3.1 システムの構造と動作

図2にシステムの全体構成を示す。システムは、階層推論機構とルールインタプリタとから成る推論機能、ルール群、形状等に関する知識、各種の基本関数群から成る知識ベース、解くべき配置計画問題の記述、部屋の記述等から成るワーキング・メモリとで構成されている。

知識ベース内のルール群は、推論制御ルール群、問題解決用ルール群をはじめとする、いくつかのルールグループに分かれている。推論制御ルール群は、ワーキング・メモリ内の状況を参照して (①)、呼び出すべきルールグループの名称を AGENDA (告知板) に登録する (②)。階層推論機構は、AGENDA に記入された情報に基づいてルールグループを呼び出す。ルールインタプリタは呼び出されたルールグループ内のルールを実行する (③)。このとき、AGENDA への新情報の書き込み (④)、ワーキング・メモリ内の情報の更新 (⑤)、各種の基本関数の実行 (⑥)、形状等の知識の参照がおこなわれる (⑦, ⑧)。

システムへの外部からの入力情報は大きく3種ある。第1はシステムへの初期知識の移植であり、専門家とシステム開発者が協力して知識ベース内に情報を登録する (⑨)。第2は通常の利用時の情報入力であり、部屋形状や配置物名称などを入力する (⑩)。第3はシステム利用時に知識を追加する場合であり、現状では配置物の位置決定操作のルールに限り、簡易入力によるルール生成が可能である (⑪)。

3.2 ルールの形式

図3にルールグループ内の個々のルールの形式を示

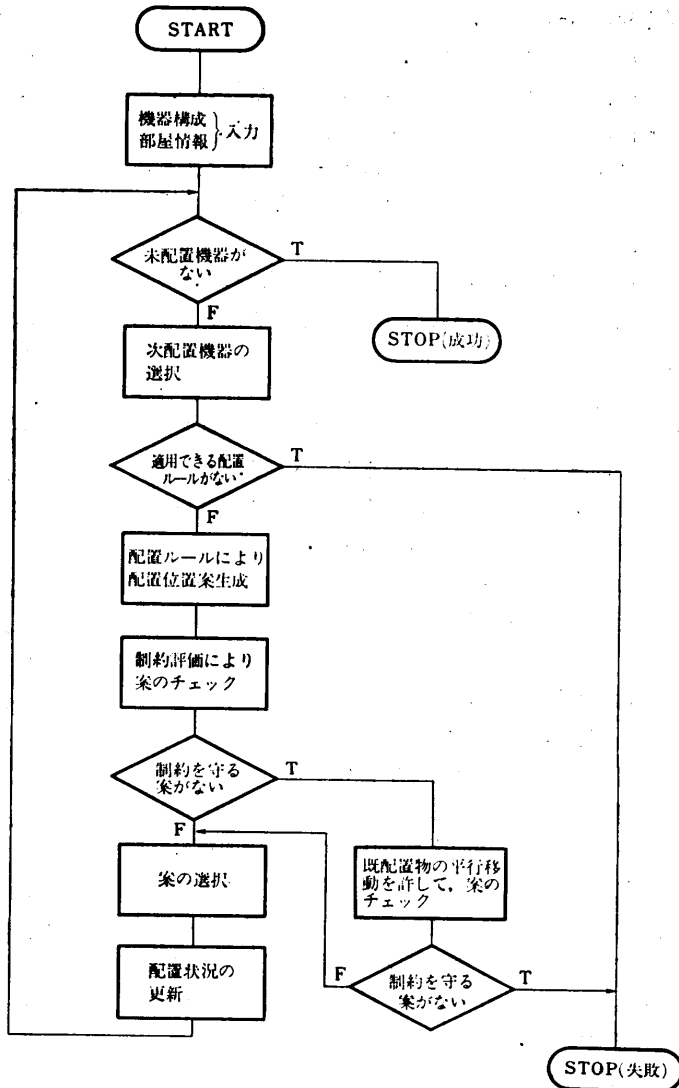


図1 配置問題解決プロセス

Fig. 1 Layout problem solving process.

した。ルールは左辺の if 部と右辺の then 部から成っている。if 部の要素 α_i は、ワーキング・メモリ内のフレームが特定の情報を保有しているか否かを確認する AOV アクション (フレーム A, スロット O, バリュ V の組みがワーキング・メモリ内に存在するか否かのチェック) およびユーザ定義関数のいずれかを使用できる。前者の場合、A, O, V に関する変数の使用が可能である。このときルールの if 部が成立しない場合、V の値については自動的にバックトラッキングする機能もある。

ルールの then 部の要素 β_j は、ワーキング・メモリ上のフレームを操作する関数、推論制御用の関数、

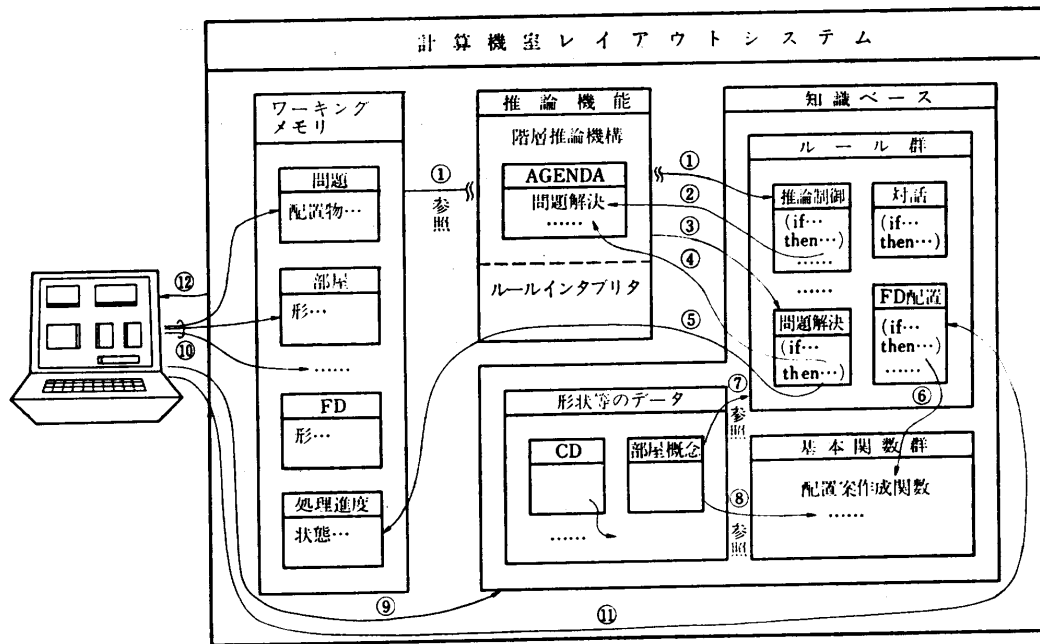
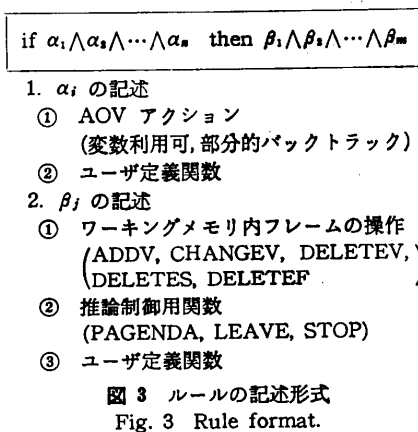


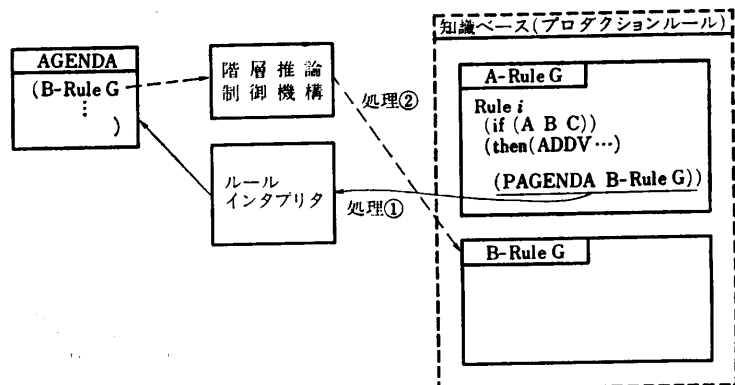
図 2 システムの全体構成
Fig. 2 System structure.



ユーザ定義関数のいずれかを使用できる。フレーム操作関数は、ADV, CHANGEV, DELETEV, DELETES, DELETEF (おのおの, フレームへのバリュの設定, 変更, 削除, スロット以下の削除, フレーム自体の削除をおこなう)から成る。推論制御関数は PAGENDA, LEAVE, STOP (おのおの, AGENDA への情報設定, ルールグループからのぬけ出し, システム動作の終了操作をおこなう) から成っている。

3.3 推論機能

図 4 に動作概要を示す。推論機能は、階層推論制御



処理 ①: A-Rule G という名のルールグループをインタプリタで処理中に PAGENDA という右辺操作関数を解釈したとき AGENDA に登録する。

処理 ②: インタプリタ処理を終了し階層推論制御機構に制御が渡ったとき AGENDA に登録されているタスク (起動すべきルールグループ名) の中から利用者指定の方法 (LIFO, FIFO 等) で、一つのタスクを取り出しこのルールグループをインタプリタする。

図 4 階層型推論機能の動作説明図

Fig. 4 Hierarchical inference mechanism.

機構と、ルールインタプリタとから成る。

階層推論制御機構は、AGENDA 上に登録されているルールグループの中から一つのルールグループを選び出す。ルールインタプリタは、選ばれたルールグループの解釈と実行をおこなう。

階層推論制御機構の処理においては、AGENDA

が空の場合、推論制御ルールグループを強制的に選択する。空でない場合には、システム起動時に利用者が設定したルール選択基準(LIFO/FIFO)に従って、AGENDA上の複数のルールグループの内の一つを選択する。

ルールインタプリタは、起動されたルールグループの解釈と実行をおこなう。グループ内のルールは先頭から順にワーキング・メモリの内容とルールの if 部がマッチングするか否かが調べられ、マッチングするものについて、右辺の then 部が実行される。以上でインプリタの処理は終了し、階層推論制御機構に制御が渡される(実際には、ルールグループごとに解釈のされ方を指定でき、指定の仕方によっては、同一ルールグループ内に、ワーキング・メモリの内容とマッチングするルールが複数存在する場合、それらをすべて実行した後、階層推論制御機構に戻すことも可能になっている。これによって、同一グループに属し、同時点で解釈・実行される機会の多いルールを同一時点で、まとめて実行させることができる)。

図5は推論機能の具体的内容を示している。①は階層推論制御およびルールインタプリタの統合管理機構の一部であり、②は階層推論制御関数、③はルールインタプリタの一部である。

3.4 ワーキング・メモリの内容

図2のワーキング・メモリ内には形状データをはじめ、各種の情報が記憶される。これらの情報を1元的に記述するために、フレームを利用した。図6(1)~(6)はワーキング・メモリの内容の一部である。これらについて以下、説明する。

(1) 問題の記述

図中の(1)は配置問題を定義したデータ構造であり、“システム DA センタ”という事物(①)は、上

```

(DEFUN NINPOON () ----- ①
  (PROG (INVRULEO OUTSTREAM INFENDFLG)
    (SETQ PRINTLEVEL 0)
    (COND ((NULL *APRULESDEV*)
      (T (SETQ OUTSTREAM
        (OUTOPEN (STREAM (ALLOO *APRULESDEV*))))))
    -----
    LOOP (COND ((NULL *AGENDA*)
      (SETQ INVRULEO 'INFERENCE-CON))
      (T
        (SETQ INVRULEO (NSETRULEG)))
      INF (COND ((NULL INFENDFLG)
        (NRINTERFRT INVRULEG)
        (GO LOOP))
      (T
        (COND ((NULL *APRULESDEV*)
          -----
          (DEFUN NSETRULEG () ----- ②
            (PROG (AM)
              (COND ((EQ *AGENDA-METHOD* 'FIFO)
                (SETQ AM (CAR (REVERSE *AGENDA*)))
                ((EQ *AGENDA-METHOD* 'LIFO)
                  (SETQ AM (CAR *AGENDA*)))
              -----
              (DEFUN NRINTERFRT (X) ----- ③
                (PROG (RGNOME HIT-STRATEGY RULELIST LEAVEFLG RULE RULE-NAME PATTERNS
                  RHS SHITFLG USED RULES RULEALL)
                  (SETQ RGNOME (CAR (GET X 'FRAME)))
                  (SETQ HIT-STRATEGY (FOET X 'HIT-STRATEGY 'VALUE))
                  (SETQ RULELIST (FOET X 'RULES 'VALUE))
                  (COND ((EQ (CAR HIT-STRATEGY) 'SINGLE)
                    (PROG ()
                      LOOP1 (COND ((OR (NULL RULELIST) SHITFLG) (RETURN NIL)))
                        (SETQ RULE (CAR RULELIST))
                        (SETQ RULE-NAME (CAR RULE))
                        (SETQ PATTERNS (OADDR RULE))
                        (PROG (FRAMESTACK SLOTSTACK VALSTACK)
                          (COND ((NLHSCON PATTERNS)
                            (SETQ RHS (ODR (OADDR RULE)))
                            (NRHSCON RHS)
                            (NAPRULESCON RGNOME RULE)
                            (SETQ SHITFLG T))
                          (T
                            (SETQ RULELIST (ODR RULELIST))
                            (GO LOOP1))))))
                    ((EQ (CAR HIT-STRATEGY) 'MULTIPLE)
                      (COND ((NULL (ODR HIT-STRATEGY))
                        -----

```

図5 推論機能の内容(部分)
Fig. 5 Inference control functions.

位概念として“SYSTEM”をもち(②)、配置すべき単位として、“L 1”以下いくつかの物体(③)をもつことを示している。部屋名称は“マシシツ1”(④)であり、物体配置に際して満足させるべき制約条件として C 1 以下の条件(⑤)があることを示している。

(2) 部屋形状の記述

図中(2)は、上記“マシシツ1”の具体的形状を定義したものである。上位概念が“マシシツ”であり、形状が“RECT”(長方形)で、たて、よこの長さが11,750×9,500(②)であること、また、部屋形状を端末に描画するときの縮尺(③)、部屋原点(④)、

```

(1) (システムDAセンサ -----①
      (AKO (VALUE (SYSTEM))) -----②
      (レイアウト (VALUE (L1) -----③
            (L2) -----③
            (L3) -----③
            -----③
            -----③
      (マシンラック (VALUE (マシンラック1))) -----④
      (レイアウト (VALUE (O1) (O2) (O3) (O4) (O5) (O6) (O7) (O8) (O9) (O10) (O11)))) -----⑤

(2) (マシンラック1 (AKO (VALUE (マシンラック1))) -----①
      (キャビネット (VALUE ((REOT 11750 9500)))) -----②
      (ハードウェア (VALUE ((+0.21800000~+01+0.21800000~+01)))) -----③
      (ソフトウェア (VALUE ((+0.10997500~+05+0.25765000~+04)))) -----④
      (ディリクタ (VALUE (ディリクタ0001))) -----⑤
      -----
      -----

(3) (L1 (AKO (VALUE (レイアウトラック))) -----①
      (キャビネット (VALUE (H-8A60-01))) -----②
      (ハードウェア (VALUE (OPU6H90))) -----③
      (ID (VALUE (OPU))) -----④
      (ルール (VALUE (OPU))) -----⑤
      (OBJ (VALUE (H-8A60-01) (OPU) (OPUレイ) (キャ) (プアマイ))) -----⑥
      (LMT-ADDED (VALUE (O1) (O2) (O6) (O7) (O8) (O10) (O11))) -----⑦
      -----
      -----

(4) (O10 (AKO (VALUE (トヨタ))) -----①
      (ディスク (VALUE (DISK) (OOP) (OPU) (PDU))) -----②
      (ディリクタ (VALUE (ディリクタ))) -----③
      (MINマモリ (VALUE (3)))) -----④

(5) (OP6 (AKO (VALUE (OP))) -----①
      (UNIT (VALUE (L1))) -----②
      (ソフトウェア (VALUE (USER))) -----③
      (IN (VALUE (マシンラック1))) -----④
      (キャビネット (VALUE (H-8A60-01))) -----⑤
      (ソフトウェア (VALUE ((SHAPE (((-500 -500) 5) (REOT 2000 2300)))))) -----⑥
      (イチ (VALUE ((10102 6417)-00))) -----⑦
      (NODE (VALUE (OP3))) -----⑧
      -----
      -----

(6) (マシンラックレイアウトレイアウト
      (ソフトウェア (VALUE (ガメン))) -----①
      (SYSTEM (VALUE (システムDAセンサ))) -----②
      (ソフトウェア (VALUE (モンダイモウリロクズ))) -----③
  
```

図6 ワーキングメモリの内容 (部分)
Fig. 6 Contents of working memory.

出入口名称 (5) 等を保持している。

(3) 配置物の記述

図中(3)は、図中(1)の配置単位の内“L 1”に関する情報であり、上位概念は“レイアウト要素”(1), L 1の機器名は“H-8 A 60-01”(2), 配置計

画時に使用する型紙すなわち物体の平面図は“CPU 6H30”(3)であり、L 1の呼び名は“CPU”である(4)。配置にあたって使用するルールグループは“CPU”ルールグループである(5)。さらに、物体の配置操作や配置時の制約条件の記述において、

“H-8 A 60-01”のみでなく，“CPU”，“CPU ケイ”，“キキ”，“ブッタイ”としても，物体 L 1 を参照可能である (6)。また，本物体 L 1 を配置した場合に考慮すべき制約条件は“C 1”から“C 11”であること (7) を示している。

(4) 制約の記述

図中(4)は，上述した制約 C 10 が，“トオク”という上位概念に属し (1)，“DICK”，“CCP”，“CPU”等の物体 (2) が，部屋の出入口 (3) から，最小距離 3 だけ離れているべきである (4) ことを示している。

(5) 物体の部屋内への配置状況の記述

図中(5)は，ある配置決定 OP 4 に関する配置情報を示したものである。図中①は，OP 4 が“OP”すなわち操作という上位概念に属しており (1)，また，OP 4 が物体“L 1” (2) に関する操作を記述

しており，部屋内の位置が“USER”から，端末を介して指定されたものであり (3)，この物体が“マシンシツ 1”内に配置してあり (4)，形は“H-8 A 60-01” (5)，保守エリア (当機器を保守する際に必要な区域) は，本体の座標原点を起点として (-800, -800) の位置に原点をもち，角度 0 で配置されており，サイズが 3,860×2,300 の長方形 (6) である。また，物体“L 1”の座標原点は，部屋内の座標系において，(10102, 6417) の点に，-90 度の方向で配置してあること (7) を示している。次の⑧は，配置計画立案過程を木探索過程とみた場合，操作 OP 4 の親ノードが“OP 3”であることを示している。

(6) 配置進捗状況の記述

図中(6)は配置操作の進捗状況を示すものであり，“システム DA センタ” (2) を端末画面に出力表示し (1)，配置計画を作成するにあたって前提となる部屋

```

(1) (INFERENCE-CON
      (HIT-STRATEGY (VALUE (SINGLE)))
      (RULES (VALUE ((I1 ((IF (マシンプレイアウトサテライト ショウタイ モンダイカイフズ)) ----- ①
                             (THEN (STOP))))
                ((I2 ((IF (マシンプレイアウトサテライト ショウタイ モンダイムウリヨクズ)) ----- ②
                             (THEN (PAGENDA YSOLVE))))
                ((I3 ((IF (マシンプレイアウトサテライト ショウタイ ガンタイギズ))
                             (THEN (PAGENDA IPROBLEM))))))
      -----
(2) (YSOLVE (HIT-STRATEGY (VALUE (SINGLE)))
      (RULES (VALUE ((S1 ((IF (モンダイカイフ ショウタイ ショウリヨク)) ----- ①
                             (THEN (DELETES モンダイカイフ ショウタイ)
                                   (DELETES モンダイカイフ, カクカク)
                                   (PAGENDA INFERENCE-CON)
                                   -----
(3) (YTOD (HIT-STRATEGY (VALUE (SINGLE)))
      (RULES (VALUE ((O0 ((IF (>X >Y >Z)) ----- ①
                             (THEN (ADDV イチギメ ソウタイイテアン USER)
                                   -----
(4) (YTFD (HIT-STRATEGY (VALUE (MULTIPLE)) (HIT-ONOBONLY)))
      (RULES (VALUE ((FD0 ((IF (>X >Y >Z))
                             (THEN (PAGENDA YLOOKATE)
                                   (CHANGEV イチギメ ショウタイ マヒヨウサアンサテライト))))
                ((FD01 ((IF (マシンプレイアウトサテライト PLAN >X)) ----- ①
                             (X OP >OD)
                             ( YIDENTIFY (OD 'OD)
                             (X OP >OR)
                             ( YIDENTIFY (OR 'OR)
                             ( YLOOK (OR
                                       (OD
                                       'ノカク
                                       'ヒダリム+(M 2)
                                       (-0.25000000へ+01 -0.50000000へ+00))))
                (THEN (ADDV イチギメ
                        ソウタイイテアン
                        ( LIST
                          (LIST (OR 'ノキナラビ ' (M 2 M))))
                        -----

```

```

(5) (DEFUN YIDENTIFY (O AKO) -----①
      (COND ((YAKOP O 'OP) (MEMBER AKO (YAKO-OP O)))
            (T (MEMBER AKO (PGETOLASSES O))))))

(DEFUN YLOOK (OP1 OP2 REL PARMS) -----②
      (FUNCALL (YIMPLODE "YLOOK-" (PNAME REL) OP1 OP2 PARMS))

-----

(DEFUN YOENTER-OP (OP) -----④
      (LETS ((TR (YFGET-V1 OP 'イテ))
            (Y+VEO (OAR TR) (YPOLAR (YMID-FRONT (YPAPER-OP OP)) (OADR TR))))))

(DEFUN YFRONT-OP (OP) -----⑤
      (LETS ((TR (YFGET-V1 OP 'イテ)) (A (REMAINDER (ABS (OADR TR)) 180))
            (COND ((EQ 0 A) (OADAR TR)) ((EQ 90 A) (OAAAR TR))))))

(DEFUN YRANGE1-FRONT (OP) -----⑥
      (LETS ((SH (YFGET-V1 (YPAPER-OP OP) 'カチ) (F (YSIDE-90 SH 0))
            (R (YSIDE 90 SH 0)) (Y (YFRONT-OP OP)) (A (OADR (YFGET-V1 OP 'イテ))))
            (COND ((OR (Y=ANGLE A 0) (Y=ANGLE A -90)) (LIST (/ -V F) (/ + V R)))
                  ((OR (Y=ANGLE A 90) (Y=ANGLE A 180)) (LIST (/ -V B) (/ + V F))))))

(DEFUN YUSER-COORDINATE (OP PLAN (NEWP T) (SLIDEP NIL)) -----⑦
      (PROG (A DEF ROOMTR VV L LMTS OBJS LMTOS OPS)
            (SETQ OBJS (YOBJLIST PLAN) OPS (YREMOVE OP (YFGET-V PLAN 'OP)))
            (COND (NEWP (SETQ LMTS (PGET-Z OP 'LMT-ADDED)
                              LMTOS (PGET-Z OP 'LMT-ADDED-OTHERS)))
                  (T (SETQ LMTS (PGET-Z OP 'LMT-CHANGED)
                              LMTOS (PGET-Z OP 'LMT-CHANGED-OTHERS))))
            (COND (SLIDEP (SETQ A (OADR (YFGET-V1 OP 'イテ))))
                  (LOOP (SETQ ROOMTR (I&ROOM (YFGET-V (YFGET-V1 PLAN 'SYSTEM) 'マシン))
                                     (COND ((NOT SLIDEP) (SETQ A (Y&AG (OAR ROOMTR) (OADR ROOMTR))))
                                     (SETQ L (YFGET-V1 (OAR ROOMTR) 'フリーアクセス))
                                     (SETQ DEF (OADR ROOMTR))
                                     (YFPUT-V OP 'IN (OAR ROOMTR))
                                     (SETQ VV (YSELEOT-SITES
                                             (LIST DEF)
                                             A
                                             (LIST (LIST (/ - (OAR DEF) L) (/ - (OADR DEF) L))
                                                  (LIST (/ + (OAR DEF) L) (/ + (OADR DEF) L)))
                                             NIL
                                             OP
                                             LMTS
                                             OBJS
                                             LMTOS
                                             OPS)))
                        (SETQ VV (YSELEOT-SITES
                                  (LIST DEF)
                                  A
                                  (LIST (LIST (/ - (OAR DEF) L) (/ - (OADR DEF) L))
                                          (LIST (/ + (OAR DEF) L) (/ + (OADR DEF) L)))
                                  NIL
                                  OP
                                  LMTS
                                  OBJS
                                  LMTOS
                                  OPS)))
            (SETQ VV (YSELEOT-SITES
                      (LIST DEF)
                      A
                      (LIST (LIST (/ - (OAR DEF) L) (/ - (OADR DEF) L))
                            (LIST (/ + (OAR DEF) L) (/ + (OADR DEF) L)))
                      NIL
                      OP
                      LMTS
                      OBJS
                      LMTOS
                      OPS)))

-----

(DEFUN YCOORDINATE (OP PLAN) -----⑧
      (LETS ((PHRS (YFGET-V1 OP 'ソウタイ) (A (YGEN-ANGLE PHRS))
            (X (YGEN-RANGE PHRS (YFGET-V1 OP 'UNIT))) (RANGE (YOAR X))
            (DEFS (YOADR X))
            (V (YSELEOT-SITES
                DEFS
                A
                RANGE
                PHRS
                OP
                (PGET-Z OP 'LMT-ADDED)
                (YOBJLIST PLAN)
                (PGET-Z OP 'LMT-ADDED-OTHERS)
                (YFGET-V PLAN 'OP))))))

```

図 7 知識ベースの内容 (部分)
Fig. 7 Contents of knowledge base.

形状や配置物の名称を入力した段階であること (③) を示している。

3.5 知識ベースの内容

知識ベース内には、形状データ、基本関数、配置操

作のためルール等の諸知識が含まれているが、これらを1元的に管理するために、基本関数以外はすべてフレームによって記述した。

(1) 形状等のデータ

部屋形状、機器諸元データ、機器型紙データ等、先に3.4節で述べたワーキング・メモリ内の諸情報のうち、問題に固有でない部分を、一般モデルとして保有している。たとえば、部屋のモデルとしては、3.4節(2)のフレーム記述の中で、VALUE フェセットの内容が未定義のものが知識ベースに記憶されている。

(2) 推論制御ルール

図7(1)は、前述した推論制御ルールグループである。たとえば①は、もしもワーキング・メモリ内の“マシンシツサクセイショリ”フレームの、“ジョウタイ”スロットに、“モンドイカイケツズミ”というデータが存在する場合には、計画作成行動を“STOP”(終了)させるという知識を表している。②は、“モンドイニューリョクズミ”となった場合に、配置計画作成処理を開始することを指示している。

図中のYSOLVEという名称のルールグループは、対話処理、自動レイアウト処理、終了処理等、当システムの動作モードを切りかえるためのものである。

(3) 配置ルール

図中(3)は、物体配置ルールの1例であり、他にマッチングするルールが見当たらないとき、“USER”モード、すなわち利用者入力によって配置計画をおこなうことを示している。

図中(4)は、物体の配置位置を指定するルール群である。そのうちの①は、カードリーダー CR と、コンソールディスプレイ CD が、特定の位置関係を満足しているとき、フロッピー・ディスク FD を、CR の右並びに、最大2単位距離の範囲に配置することを試みよということを表現している。

(4) 基本関数

上述した各種のルールの解釈と実行をおこなうために多くの基本関数が用意されている。たとえば、各種データの外部ファイルと本システムの間での転送処理用の関数、利用者との対話処理用の関数、図形出力関数、配置ルール上の“右、左”等の語を物理座標に変換する関数、形状物の重なり状況等を評価チェックする関数、配置が不可能な場合に既配置物を移動操作する関数などがその一例である。

図中(5)の内の①～⑧はこれらの関数の一部であり、たとえば⑧は配置ルールによって指示される場所を物理座標値に変換する関数である。

3.6 ルールエディタ

図2内のルールグループのうちの物体の配置位置案

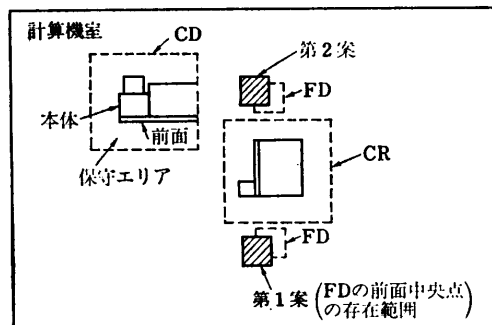


図8 配置ルールの簡易生成
Fig. 8 Placement rule generation.

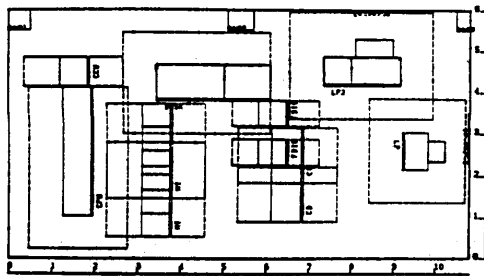
を指定するものは、if部に既配置物の相対的位置関係をthen部に配置可能範囲を指定するデータを保有している。図7(4)の例は、フロッピーディスクFDに関するルールで、その意味は“FDを配置するとき、もしカードリーダーCRが、コンソールディスプレイ装置CDの右方・左向きにある距離範囲にあれば、第1案としてFDは、カードリーダーCRの右並び、第2案はCRの右方・左向きの位置に配置せよということを示している。図8はこの意味を図示したものであるが、様々の配置状況に対して図7(4)の形のルールを手で知識ベースに入力することは大変労力を要する。そこで、配置ルールを簡単に生成することのできるエディタを作成した。エディタの動作は下記のとおりである。

(1) 利用者対話モードの中でルールエディタを呼び出すと、図2内の基本関数内のルールエディタが実行される。このとき、画面にレイアウト図が出ている必要がある。

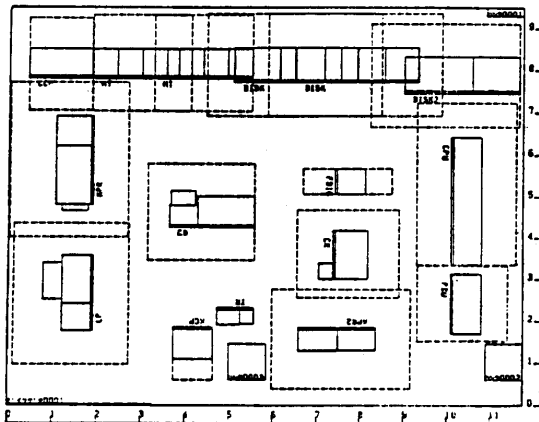
(2) ルールエディタは、利用者に配置機器名(上例ではFD)、基準機器名(CD)、既配置物名(CR)を聞く。利用者は質問への応答を、画面上の既配置機器をカーソル指定することで応答する。エディタはカーソル指定点に存在する機器名を内部で認識する。

(3) 次に、ルールエディタは、ルールの右辺を作成するために、配置機器(FD)の配置可能位置を質問する。利用者はカーソル入力により質問に答える。複数の場所を指定できる。

(4) ルールエディタは、以上の入力を得て、図7(4)の形式のルールを作成する。この際、数値入力データを“ミギナラビ”、“ヒダリナラビ”等のルール記述語に自動変換する。作成したルールは、知識ベース内に格納される。



(1)



(2)

図9 実験結果出力
Fig. 9 System outputs.

4. 実 験

(1) システムの開発

上記のシステムを M 180 VOS 3 上で UTILISP 言語上に開発した。プログラム規模は約 6k ライン、推論制御機能はそのうち約 1k ラインである。知識ベース内のルール数は現状約 100、フレームの数約 100 程度である。

(2) レイアウト計画の作成

開発したシステムは、社内で計算機販売に従事するシステムエンジニアの業務支援用に実用されている。従来、1件につき5~8時間を要していたものが、30分程度で作成可能となった。

図9(1),(2)は本システムによって部屋形状および配置物の異なるケースに対する配置計画を作成させたものである。いずれのケースにおいても、2.1節で述べた制約を満足する配置計画を作成できている。

また、3.6節のルールエディタによって、従来ルール1件の作成に1時間程度必要であったものが、数分

程度で実施できるようになった。

5. ま と め

5.1 結 論

計算機室への機器配置問題を解くための知識ベースシステムを開発し実用化した。本システムは前向き型の階層推論機能と、配置計画作成に使用する形状知識、配置ルール等から成る知識ベースをもち、M 180 VOS 3 上に UTILISP を用いて、約 6k ラインで実現されている。一部のルール(配置ルール)については、配置機器や基準機器の名称および配置可能範囲をカーソル入力することにより、S式表現されたルールを生成する機能をもっている。

5.2 今後の課題

現在のシステムには、比較的小数の知識しか入っていない。今後、数多くのレイアウト問題を処理させる中で、知識の充実を図りたい。

謝辞 本システムは昭和57年7月から58年5月の期間で開発したものであるが、開発に際して UTILISP を利用させていただいた東京大学工学部・和田英一教授、近山隆氏に深謝します。また、本研究の機会を与えてくださった日立製作所システム開発研究所、川崎淳所長、井原広一副所長、春名公一部長に、また同ソフトウェア工場、金子雄次専門コンサルタント室長、竹下仁志主任技師、竹沢勝雄主任技師に深謝します。開発にあたって推論技法等に関する情報を提供していただいた千吉良英毅研究員、関主管研究員に深謝します。

参 考 文 献

- 1) Liggett, R.S.: Optimal Space Planning in Practice, CAD, Vol. 13, No. 5, pp. 277-288 (1981).
- 2) Pfefferkorn, C.E.: A Heuristic Problem Solving Design System for Equipment of Furniture Layouts, C. ACM, Vol. 18, No. 5, pp. 286-297 (1975).
- 3) McDermott, J.: R1: A Rule-Based Configurer of Computer Systems, CMU-CS-80-119.
- 4) Roth, B.H., Roth, F.H. et al.: Modelling Planning as an Incremental, Opportunistic Process, Proc., IJICAI, Tokyo, pp. 375-383 (1979).

(昭和59年6月29日受付)

(昭和60年3月20日採録)