

問題と事例の分割に基づいて部分的類似例を利用する 事例ベース推論方式†

吉 浦 裕††

エキスパートシステムの構築において知識獲得の難しさが問題になっている。その一つの原因は、経験的知識をルールや手続きに変換することの難しさにある。これに対し、過去の事例に基づいて問題を解く事例ベース推論によれば、知識の多くを事例の形で入力できるので、上記変換作業を軽減できるという期待がある。本論文では、知識獲得の容易化を目的とした、事例以外の知識をできるだけ必要としない事例ベース推論方式について述べる。事例ベース推論では、問題に該当する事例がない場合のために、事例以外の問題解決知識が必要になる。そこで、問題および事例を分割し、部分問題ごとに該当する部分事例を見出し、そこから得た部分解を組み合わせて全体解を求める方式により、事例以外の知識の量を削減した。ところが、多くの分野では、部分問題間の干渉のため、全体の正しさと局所的正しさが一致しない。そのため、事例が全体として正しくても、その中の利用部分が正しいとは限らず、利用部分が正しくても、それを複数組み合わせて問題全体の正解が得られるとは限らない。そこで、部分問題間の干渉について、現在の問題解決で起こるものと、事例の解決過程で起こったものを解析することにより、問題と事例の適切な分割を求める方式を開発した。本方式を用いて計算機室レイアウトシステムを開発し、ルール型推論を用いた同機能のシステムとの知識の比較により、知識獲得容易化の見通しを得た。

1. はじめに

近年、エキスパートシステムが、製造業から金融、流通業など幅広い分野で実用化されるようになってきた。エキスパートシステムを構築するためには対象分野の知識を計算機に入力する必要がある。システムの必要とする知識の内容、表現形式は、そのシステムの推論方式に依存する。従来のエキスパートシステムの多くは、ルール型推論方式に基づいており、ここでは、専門家の経験的知識をルール形式で入力する必要があった^{1),2)}。

ところが、これらの知識は専門家が無意識に用いたり、あいまいであったりするため、明確に整理することが難しいものが多く、また、明確化できるものでも、ルール形式での記述に適さないものがある。そのため、知識獲得がエキスパートシステム構築の重大な障害となっている^{3),4)}。

この問題の一つの解決策としては、知識獲得の方法論および支援環境を開発することである。例えば、知識獲得作業のモデル化³⁾、知識のタイプに依存したインタビュー技術^{5),6)}、専門家向けの分かりやすい知識表現を提供し、これを内部でルールに変換する方法⁶⁾、ルールの記述を誘導し、その構文および意味をチェッ

クする知的エディタをアプリケーションごとに設ける方法⁷⁾、ルール群を解析してルールの不足や重複を指摘する方法^{7),8)}などが研究されている。

知識獲得を容易化する他の方法としては、ルール型以外の推論方式を採用することにより、必要な知識の内容、表現を獲得に適したものにすることが考えられる。我々は、そのような方法の一つとして、過去の事例に基づいて問題を解く事例ベース推論⁹⁾⁻¹⁵⁾を取り上げた。

事例ベース推論を用いてエキスパートシステムを構築すれば、知識の多くを事例の形で入力できる可能性がある。事例は、以下の理由から、ルールに比べ獲得が容易であると思われる。

(1) 知識が無意識に用いられていたり、あいまいであっても、その適用結果である事例は明確な事実である。

(2) ルールが文書化されている分野は多くないが、事例が文書化あるいはデータベース化されている分野は多い。

本論文では、知識獲得の容易化に重点をおいた事例ベース推論の方式を述べる。例題として、計算機室に各種の機器を配置する問題(計算機室レイアウト問題)を用いる。

以下、2章では推論方式の設計方針、3章では計算機室レイアウト問題における具体的な推論方式を述べる。4章では、本方式による計算機室レイアウトシステムと、ルール型推論を用いた同機能のシステム¹⁶⁾を

† A Case-Based Reasoning Method Using Partially Similar Cases by Decomposing Problem and Cases by HIROSHI YOSHIURA (5th Department, Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所第5部

比較することにより知識獲得容易化の評価を行う。

2. 知識獲得の容易化に重点をおいた事例ベース推論の設計方針

2.1 計算機室レイアウト問題

推論方式の説明に先立って、例題の簡単な説明を行う。計算機室レイアウト問題とは、中央処理装置 (CPU)、コンソールドisplay (CD) などの機器を、計算機室内に配置する問題である。配置にあたっては、利用者の使い勝手、保守の容易性、見栄えなどの理由から、下記のような様々な制約を満たす必要がある¹⁶⁾。

(1) カードリーダー (CR)、フロッピディスクユニット (FDU)、ラインプリンタ (LP) などの入出力機器は、CD の前に座るオペレータが利用しやすいように、CD の近くに置く。

(2) LP、漢字プリンタ (KPR) は、室外から来る利用者のために、出入り口の近くに置く。

(3) 機器の種類ごとに、定まった保守エリアを確保する。

これらの制約を満たすレイアウトを作成するには、通常1日程度、熟練した SE でも3時間程度要する。そこで、SE 業務の効率化の点から、自動レイアウトに対するニーズがある。

この問題に対する一つのアプローチとしては、ルール型推論に基づくシステムがある。下記は、実際のルール型システム¹⁶⁾で使われている FDU の配置ルールを、日本語表現で表したものである。

「もし、CD と CR が既に配置されていて、
CR が CD の右方、2m 以内、左向きで
CD に対する前後のずれが
前方 0.5m から 2.5m の間であり、
CR の右並び 2m 以内にスペースがある
ならば、CR の右並び 2m 以内、なるべく近くに、
CR と同じ向きに、FDU を置け。」

ルール型推論を用いた計算機室レイアウトシステムでは、このような複雑なルールが数百から数千個必要であり、知識獲得に大きな手間を要していた。

一方、計算機メーカーでは、計算機室レイアウトの膨大な事例が、CAD データベースなどの形で既に電子化されており、容易に利用できる状況にある。また、CPU を中小型機種に限れば、配置される機器の種類および部屋の形状のバリエーションは限定される。す

なわち、問題に対して十分な事例が利用可能である。

ここでの例題は、部屋と機器からなる問題に対して、上記のようなルールの代わりに、過去の事例に基づいて、レイアウトを生成することである。

2.2 知識獲得容易化のための設計方針

本研究の目的は、事例の形で知識を入力することにより、その獲得を容易化することである。したがって、事例以外の知識をなるべく必要としない事例ベース推論方式を目指すことにする。具体的には、以下の方針に従った。

(1) 事例の解の再利用

従来の事例ベース推論システムの多くは、現在の問題とは異なる問題の解決事例から、解を類推している^{9)-12),14)}。例えば、料理の計画システム CHEF^{10),11)} は、鶏肉と豆のいため料理に関する事例から、牛肉とブロッコリのいため料理の計画を類推する。

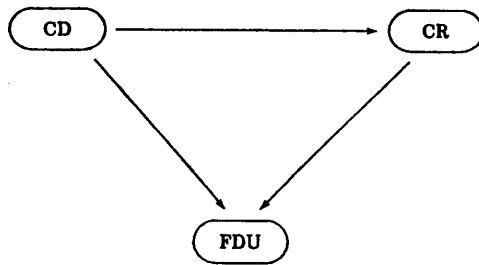
ところが、実用に耐える精度の類推システムを構築するには、二つの事物の間の類推の妥当性を検証する知識や、事物間の細かい相違に関する知識など、従来のルール型推論では必要でなかった知識が新たに必要となる。したがって、この方式は知識獲得容易化という目的に対して、最初に試みるべき解決策ではないと思われる。そこで、現在の問題あるいはその部分問題と同じ事例を検索し、そこでの解を再利用することにした。

(2) 部分問題間の干渉の回避

一つの事例だけでは問題全体を解決できない場合がある¹⁰⁾⁻¹⁴⁾。そこで、事例を用いて問題の一部を解決し、問題の残り部分を何らかの方法で解決して、それぞれの部分解から全体の解を求めることが必要である。このとき、最初に事例を用いて解いた部分問題と後で解いた部分問題の間の干渉が問題になる。

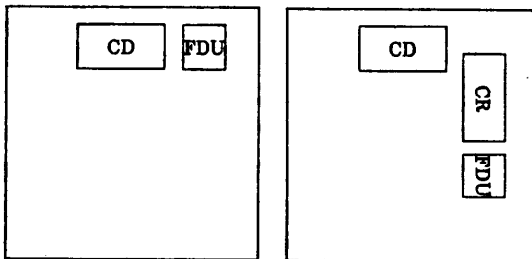
部分問題間の干渉とは、一つの部分問題の解が他の部分問題の存在により影響を受けることである。部分問題間に干渉がある場合、部分問題を単独で解いた場合の解が、問題全体の解の一部としてそのまま利用できるとは限らない。したがって、事例を用いて解いた部分問題の解が、そのまま利用できるとは限らない。

例えば、計算機室レイアウト問題では、指定された機器全部の配置が全体問題であり、一部の機器あるいは個々の機器の配置が、その部分問題である。ここでは、機器の配置の間に、機器 A の配置は機器 B の配置結果に依存するといった関係がある。この依存関係は、部分問題間の干渉の一種と考えられる。



注) ①→②は、Yの配置がXの配置結果に依存することを表す。

図1 機器の配置間の依存関係の例
Fig. 1 Examples of dependency relationship between locations of apparatuses.



(a) CRがない場合のFDUの配置
(a) Location of FDU without CR.
(b) CRがある場合のFDUの配置
(b) Location of FDU with CR.

図2 機器の配置間の依存関係の影響
Fig. 2 Influence of dependency relationship between locations of apparatuses.

例えば、CD, CR, FDUの配置には、図1のような依存関係がある。そこで、CD, FDUを配置する問題では、FDUの配置は、CDの配置結果を基準に決定され、図2(a)のようになる。ところが、CD, CR, FDUを配置する問題では、FDUの配置は、CD, CRの両者を基準に決められ、図2(b)のようになる。そのため、CD, CR, FDUの配置問題において、図2(a)のような配置事例を用いてCD, FDUの配置を求めても、これをそのまま利用して全体の配置(図2(b))を求めることはできない。

この問題に対して、以下の解決策が考えられる。

(a) 干渉を考慮せずに事例から解を求めた後、これを修正する^{10),11)}。例えば、CD, CR, FDUの配置問題において、図2(a)の事例を用いて、CD, FDUを配置した後、FDUの配置を図2(b)のように修正する。

(b) 干渉を回避する^{12),14)}。すなわち、事例を用いて求めた部分問題の解が、そのまま利用できるように、部分問題の切り出し方を制御する。例えば、上記

の例でFDUの配置を修正しなければならない原因は、FDUがCRに依存するにもかかわらず、CRを含まない部分問題を切り出したことにある。そこで、依存関係を解析することにより、修正が不要な部分問題を切り出す。

部分問題間の干渉については様々な研究が行われているが、未だその規則性は明らかにされていない¹⁷⁾。そのため、(a)では、あらゆる干渉を考慮した修正知識を設ける必要がある。例えば、図2(a)におけるFDUの配置を同(b)のように修正せよといった知識を、機器間依存関係のあらゆる影響について用意する必要がある。これは、知識獲得容易化の目的に反する。そこで、(b)を採用した。

(3) 事例ベース推論の再帰的適用

前述のように、一個の事例では問題全体が解決できない場合があるため、残った部分問題の解決方法が必要である。その方法としては、ルール型推論による方法^{10),11)}と事例ベース推論を再適用する方法¹²⁾⁻¹⁴⁾がある。

どの部分問題が残るかは、問題と事例の組合せに依存し、あらかじめ決められないため、ルール型推論を用いた場合、様々な部分問題の解決ルールを用意する必要がある。これは、知識獲得容易化の目的に反する。そこで、事例ベース推論を再適用することにした。

2.3 従来の事例ベース推論方式の問題点とその解決方針

2.2節(2)で述べたように、本方式では、事例を用いて解いた部分問題の解が、問題全体の解の一部としてそのまま利用できるように、部分問題の切り出し方を制御する。そのような事例ベース推論方式の例としてはMEDIATOR^{12),14)}があるが、そこでは、部分問題およびその解決順序があらかじめプログラムされている。

ところが、3.1節(5)で述べるように、部分問題ごとに異なる事例に基づいて解くと、解全体の一貫性が失われることがある。そこで、問題分割をできるだけ避けるために、一つの事例を用いて、できるだけ大きな部分問題を解きたいという要求がある。例えば、一つの事例からできるだけ多くの機器を配置したい。そこで、本方式では、前記の機器間の依存関係に基づいて問題を解析することにより、問題と事例の組合せに応じて、問題の中から、事例を用いて干渉の補正なしに解決可能な最大の部分問題を、動的に切り出す。

一方、部分問題間の干渉は、問題分割だけではなく、事例の解から必要部分を切り出す時にも考慮する必要がある。事例の解が、部分問題間の干渉の結果得られた解である場合には、その一部を取り出して、干渉のない、あるいは異なる干渉を含む問題の部分解として、そのまま利用することはできない。

例えば、図2(b)のような事例から、CDとFDUのみの配置を切り出しても、適切な配置(図2(a))は得られない。事例からの情報抽出において部分問題間の干渉に対処することは、従来研究では、ほとんど考慮されていない。本方式では、現在解きたい問題だけでなく、事例の問題についても依存関係に基づく解析を行い、修正の必要がない部分解を切り出す。

以上の依存関係に基づく解決策については、3.2節で具体的方式^{18),19)}を述べる。

3. 計算機室レイアウト問題における本方式の実現

3.1 概要

本章では、前章の方針に従って計算機室レイアウト問題の分野でインプリメントした事例ベース推論システムについて述べる。ここでは、問題を、CPUが単一かつ中小機種、部屋の形が長方形で近似できるものに限定した。図3に示すように、本システムにおける

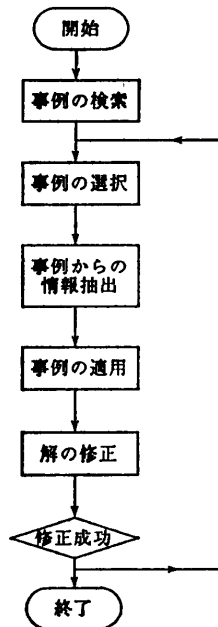


図3 計算機室レイアウト問題における本方式の基本処理

Fig. 3 Main processes of the present method for computer room layout.

事例ベース推論の基本処理は、事例の検索、事例の選択、事例からの情報抽出、事例の適用、解の修正から成る。以下、それぞれの処理を説明する。

(1) 事例の検索

事例の全体集合から、現在の問題に類似した事例の集合を切り出す。最適事例の選択は、本処理ではなく次の事例選択処理で行う。問題と事例の類似性の定義は、機器の種類類似性と、部屋の形状の類似性に基づく。具体的な処理手順は以下のとおりである。

(a) 現在の問題との共通機器が多い順に、指定された個数だけ事例を検索する。

(b) 検索した事例を、部屋の縦辺および横辺の長さ、出入口の位置の三つの基準について、現在の問題と比較し、すべての基準について、問題と事例の距離が指定された許容値以内であれば、事例を採択し、そうでない事例を除外する。

(c) 上記三つの基準から成る空間を想定し、この三次元空間における問題と事例の距離が小さいものから、指定された個数だけ事例を採択する。

上記のような、問題と事例の特徴集合の共通度に基づく選択、特徴の相違に基づく除外、特徴の近似度に基づく優先順位付けを組み合わせた事例検索方式は、従来の事例ベース推論で一般的に採用されている¹⁰⁾⁻¹⁴⁾。

(2) 事例の選択

検索した事例の中から、問題解決に利用できる機器を最も多く含んだ事例を選択する。ここで、利用可能な機器とは、問題と事例の共通機器のうち、2.2節(2)および2.3節で述べたように、部分問題間の干渉を補正せずに、配置をそのまま現在の問題に適用できるものである。利用可能機器の切り出し方法については、次節3.2で詳述する。

(3) 事例からの情報抽出

選択された事例から利用すべき情報を抽出する。2.1節で述べた計算機室レイアウトに関する制約は、機器および出入口の間の相対的な位置と向き(以下、相対位置と略す)に帰着される。そこで、前述の利用可能な機器および出入口の間の相対位置を抽出する。

(4) 事例の適用

抽出した相対位置を現在の問題に適用することにより、機器を配置する。具体的には、事例の出入口が現在の問題の出入口に一致するように、事例の機器を現在の部屋に配置する。

(5) 解の修正

事例適用により配置された機器の位置の修正と、ま

だ配置されていない機器（未配置機器）の配置を行う。以下、それぞれについて述べる。

(a) 機器の位置の修正

上記事例検索の(b)における部屋形状の許容誤差が経験的であるため、現在の問題にそのまま適用できない事例が検索されることがある。その場合、機器が部屋からはみだすという問題が生じる。同様に、事例の検索時に、部屋の形状を長方形に近似したり、柱を無視するなどの単純化を行っているため、その適用時に、機器が部屋からはみだしたり、柱と重なるという問題が生じる。

本処理では、指定された許容範囲内で機器の位置を修正することにより、これらの問題を解決する。ここでは、ある配置が妥当ならば、その中の機器の位置を微修正した配置も妥当である、という経験則を用いている。

位置の修正には、ルール型推論を用いる。許容範囲内の位置修正で上記問題が解決できない場合には、事例選択処理に戻り、異なる事例に基づいて配置を試みる。

(b) 未配置機器の配置

本処理は、一般論で言えば、一つの事例を用いても解決できなかった部分問題を解決することに相当する。ここでは、2.2節(3)で述べたように、事例ベース推論を再適用する。処理手順は、以下のとおりである。

①部分問題の生成

機器の配置の間の依存関係に基づいて、未配置機器が、すでに配置された機器(既配置機器)に依存しているかどうかを判定する。もし、依存している場合には、未配置機器とそれが依存している既配置機器の配置問題を、新しい部分問題とする。依存していない場合には、未配置機器のみの配置を新しい部分問題とする。

②部分問題に対する事例ベース推論の実行

上記の部分問題に対して、本事例ベース推論を再適用する。ただし、新しい部分問題が既配置機器を含む場合には、既配置機器と未配置機器の相対位置を事例から抽出し、出入口の代わりに既配置機器を基準として、未配置機器を配置する。

上記のうち、未配置機器が既配置機器に依存する場合の処理は、機器間の相対位置が、部屋の形状や他の機器の存在にかかわらず一定である場合が多いという経験則を用いている。

この事例ベース推論の再適用は、二つの理由から解の質を低下させる。一つは、上記経験則の不確実性に

よる。もう一つは、計算機室レイアウト問題は解決方法が専門家によって少しずつ異なり、また、同じ専門家でも時間の経過によって異なる点である。そのため、複数の事例の解を組み合わせると、全体として一貫性のない解になる恐れがある。

これらの理由のうち、前者は、本方式に特有であるが、後者は、問題解決方法が完全に定まっていない分野一般にあてはまる。

3.2 事例からの利用可能部分の切り出し

本節では、事例の解から、部分問題間の干渉を補正せずに、そのまま現在の問題に適用できる部分を切り出すための方式^{18),19)}を述べる。ここでは、2.3節で述べたように、現在の問題と事例の問題の両者について、部分問題間の干渉を考慮する。

(1) 用語の定義

まず、概念および用語を明確化する。

(a) 部分問題および共通部分問題

問題Pの解を求める過程で問題P'が解決されるときに、P'をPの部分問題であるという。問題P'が、問題P1とP2の両者の部分問題であるとき、P'をP1とP2の共通部分問題という。

(b) 部分解

問題P'が問題Pの部分問題であるとする。このとき、Pの解を求める過程で得られたP'の解を、Pの解決過程におけるP'の解という。問題Pの解決過程で得られた部分問題の解を、単に、問題Pの部分解ということもある。

(c) 部分問題間の干渉

問題Pの二つの部分問題P1, P2が次の条件を満たすことを、PにおいてP1がP2に干渉するという。

「問題Pの解決過程で、部分問題P1を変化させたときに、部分問題P2の解が変化することがある」

部分問題P1を変化させるとは、Pの解決過程におけるP1と解と副作用のうち少なくとも一方を変化させることである。これは、P1の解と副作用を空にする場合、すなわち、P1を除く場合も含む。また、P2の解の変化は、P2の解が空になる、すなわち、P2が解決不能になる場合も含む。

ここで扱っている計算機室レイアウト問題では、ある機器集合を配置する問題に対して、その一部の機器を配置することが部分問題であり、その配置結果が部分解である。

上記(c)で述べたように、部分問題間の干渉は、そ

これらの部分問題を含む全体問題に依存する。そのため、干渉解析には文脈に依存した複雑な知識が必要となる。例えば、「通信制御装置の配置は、CPU の配置を含まない問題（例えば、CPU を別室に配置する場合）では DKU の配置に依存するが、CPU と DKU の両者を含む問題では、DKU ではなく CPU に依存する」といった知識が必要である。そこで、干渉関係を厳密に解析することは知識獲得の容易化に反すると考え、以下の概念を導入した。

(d) 部分問題間の干渉可能性

P において P1 が P2 に干渉するような P が、対象とする問題分野に存在することを、P1 が P2 に干渉可能であると呼ぶ。

干渉可能関係は、全体問題には依存しない。そこで、その解析に必要な知識は、「通信制御装置の配置は DKU に依存する可能性がある」、「通信制御装置の配置は CPU に依存する可能性がある」といった独立した二項関係であり、干渉関係の知識に比べて獲得が容易であると思われる。2.2 節(2)で述べた機器の配置間の依存関係は、部分問題間の干渉可能関係に相当する。

本方式では、干渉可能関係に基づいて利用部分を切り出す。すなわち、部分問題間の干渉可能性を回避するように部分解を切り出す。干渉と干渉可能性の関係から明らかのように干渉可能性の回避によると、干渉の回避よりも厳しい条件で利用可能部分を切り出すことになる。

(2) 利用可能部分の満たすべき条件

前記 2.2, 2.3 節の方針より、利用可能部分は、事例の解のうち以下の条件の積を満たす部分である。ただし、利用可能部分を AP (Available Part) と略すことにする。

(a) AP は、現在の問題と事例の問題の共通部分問題 Pc の、事例の問題の解決過程における解である (2.2 節(1)より)。

(b) AP は、現在の問題における部分問題間の干渉を補正しなくても、現在の問題の解決過程における Pc の解として利用できる (2.2 節(2))。

(c) AP は、事例の問題における部分問題間の干渉を補正しなくても、現在の問題の解決過程における Pc の解として利用できる (2.3 節)。

ここで、部分問題間の干渉可能関係に基づいて利用可能部分を切り出すために、上記の (b), (c) を干渉可能性に関する条件で置き換える。付録において証明

するように、 $(a) \wedge (b) \wedge (c)$ が成り立つための十分条件は、以下の (a), (b1), (c1) の積である。

(a) 上記と同じ。

(b1) 現在の問題のうち Pc 以外の部分問題が、Pc に干渉可能でない。

(c1) 事例の問題のうち Pc 以外の部分問題が、Pc に干渉可能でない。

本方式では、条件 $(a) \wedge (b1) \wedge (c1)$ に基づいて、利用部分を切り出す。この条件が、上記 $(a) \wedge (b) \wedge (c)$ の十分条件であるため、所期の性質を持った部分解を切り出すことができる (ただし、必要条件ではないため、所期の性質を持った部分解全体ではなくその一部が切り出される場合がある)。

(3) 利用可能部分の切り出し方式

上記条件を満たす部分を切り出す具体的方式は、問題領域および問題解決モデルに依存する。ここでは、計算機室レイアウトシステムにおいて我々がインプリメントした方式を述べる。本方式は、部分問題間の干渉可能性に関する知識として、機器間の依存関係を用いる。処理手順は、以下のとおりである。

(ステップ1) 現在の問題および事例に含まれる機器間の依存関係の有無を決定する。依存関係には明らかに推移律が成り立つ (機器 A の配置が、機器 B の配置結果に依存し、B が C に依存するならば、A が C に依存する) ので、機器間の直接の依存関係以

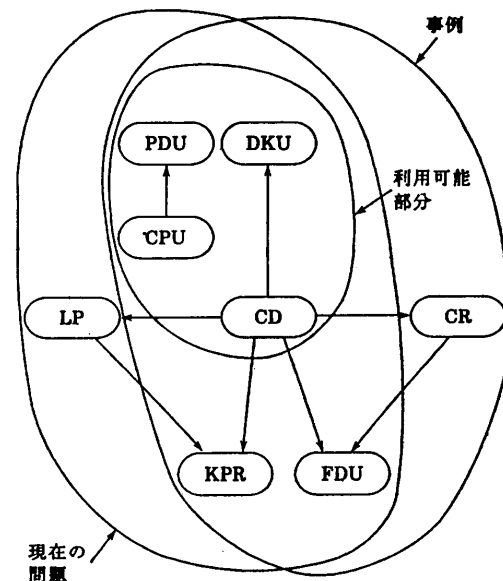


図4 依存関係に基づく利用可能部分の切り出しの例
Fig. 4 Example of extraction of usable part based on dependency relationship.

外に、間接的な依存関係も求める。

(ステップ2) 問題と事例に共通する機器集合を求める。この集合が、上記条件(a)を満たす共通部分問題に相当する。

(ステップ3) 上記機器集合のうち、非共通機器(問題と事例の一方にのみ含まれる機器)に依存しないものの集合を求める。これが、条件(a)を満たす部分問題のうち、(b1)、(c1)を満たす部分に相当する。

(ステップ4) ステップ3で求めた機器集合の配置を事例から切り出す。これが利用可能部分となる。

図4は、CPU、CD、DKU、CR、FDU、KPR、電源装置(PDU)の配置事例を用いて、CPU、CD、DKU、FDU、LP、KPR、PDUの配置問題を解く場合の、利用可能部分を示す。

4. 知識獲得容易化に関する評価

我々は、上記の方式を用いて計算機室レイアウト用の実験システムをインプリメントした。本システムは、Prolog 約7kステップで記述され、ワークステーション上で稼働している。本システムの出力例を図5に示す。処理時間は、図5程度の規模のレイアウトで約30秒である。ここでは、本システムとルール型推論を用いた同機能の実験システム¹⁶⁾の知識を比較することにより、本方式による知識獲得容易化の効果を評価する。

本システムは、ルール型の実験システムと同等の機能を目指して開発した。ルール型システムと同じクラスの問題、すなわち、CPUが単一かつ中小機種、部屋の形が長方形で近似できる小規模の問題を対象とした。レイアウトの質がルール型と同等になるまで、知識の拡張と評価実験を繰り返した。

表1は、本システムとルール型システムの知識を比較したものである。本システムの知識のうちレイアウト事例は、2.1節で述べたように、すでに多数がDB化されている。したがって、本システムをDBに結合すれば、これらを利用できる(ただし、現時点ではDBとの結合は実現しておらず、評価実験に必要な事例を手手で移植した)。機器間の依存関係は、配置ルールに比べ、項目数、各項目の情報量共に少ない。部屋の形状に関する許容値、機器位置の修正ルールは共に小量である。以上から、計算機室レイアウト問題に関し、実験レベルでは、本方式による知識獲得容易化の効果があると考えられる。

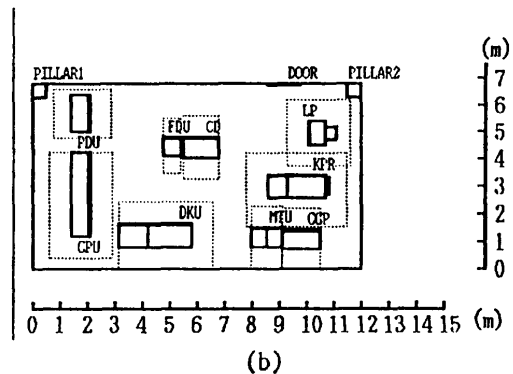
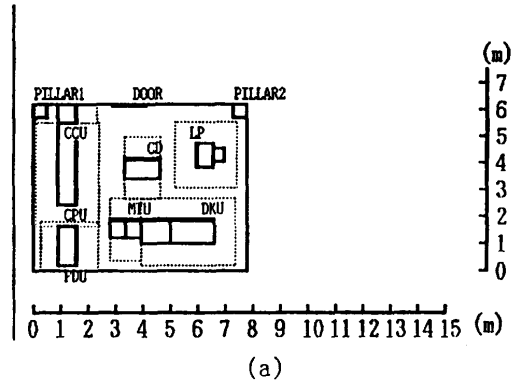


図5 本システムの出力例
Fig. 5 Examples of system output.

表1 ルール型システムと本システムの知識の比較
Table 1 Comparison of knowledge used by rule-based system and that used by the present system.

ルール型システム	本システム
配置ルール (90項目) 例) 「もし、 CDとCRがすでに 配置されていて、 CRがCDの右方、 2m以内、左向きで CDに対する前後の ずれが 前方0.5mから2.5m の間であり、 CRの右並び2m以 内にスペースがある ならば、 CRの右並び2m以 内、なるべく近くに CRと同じ向きに、 FDUを置け」	レイアウト事例 機器間の依存関係 (30項目) 例) 「FDUの配置は、CDの配置に依存する」 部屋の形状に関する許容値 (5項目) 例) 「類似例の部屋の縦の長さは、問題の部屋 ±2m 以内であること」 機器位置の修正ルール (20項目) 例) 「もし、機器が部屋からはみ出しているならば、それを、部屋の内側にはみ出した距離だけ移動せよ」
機器の形状など	同左

5. おわりに

エキスパートシステムの知識獲得を容易化するため

に、事例以外の知識をできるだけ必要としない事例ベース推論方式を開発した。

本方式では、事例の解を現在の問題に適合させるための知識をできるだけ不要化するために、現在の問題にはほぼ等しい問題あるいはそれを包含する問題の解決事例を検索し、その解を再利用する。そのような事例が存在しない場合には、問題を分割し、部分問題ごとに事例を適用する。

問題を分割して解く場合、部分問題間の干渉のため、解全体の正しさと部分解の正しさが一致するとは限らない。そのため、事例の解が全体として正しくても、その中の利用部分が正しいとは限らず、利用部分が正しくても、これを現在の問題の部分解として用いて正しい全体解が得られるとは限らない。そこで、現在の問題および事例の問題について部分問題間の干渉を解析することにより、事例の解のうち現在の問題にそのまま適用できる部分を切り出す方式を開発した。その結果、干渉を補正するための知識が不要になった。

本方式をインプリメントした計算機室レイアウトシステムと、ルール型推論に基づく同機能のシステムの必要知識を比較し、計算機室レイアウト分野における知識獲得容易化の見通しを得た。

本方式の特徴を整理すると、以下のようになる。

(1) 問題と事例の組合せに応じて、問題の中から、事例を用いて解決できる最大の部分問題を動的に切り出すことができる。その結果、一つの事例を用いてできるだけ大きな部分問題を解くことが可能になり、多数の事例の解を寄せ集めることによる解の質の低下を防止できる。

(2) 現在の問題における部分問題間の干渉を解析するだけでなく、事例の問題解決過程で起こった干渉を推定する。その結果、後者の干渉に対する補正も不要化できる。

(3) 本方式のうち、事例中の利用可能部分および問題中の解決可能部分の切り出し方法は、部分問題間の干渉関係に関する知識のみを利用するので、干渉関係が分析可能な分野一般に適用できる。

謝辞 本方式を用いた計算機室レイアウトシステムをインプリメントしていただき、研究内容に関し御討論いただいた日立西部ソフトウェア(株)橋本和広氏、本研究を御指導いただいた(株)日立製作所システム開発研究所第5部部长森文彦博士、計算機室レイアウト問題に関し御教授いただき、レイアウト事例を提供いただいた同社情報システム工場山中止志郎氏、ルール

型推論を用いた計算機室レイアウトシステムに関する研究成果を参考にさせていただいたシステム開発研究所渡辺俊典博士、安信千津子氏、研究内容に関し御指導いただいた同研究所広瀬正氏、辻洋氏各位に感謝いたします。本研究の機会を与えていただいた同研究所堂免信義所長に感謝いたします。

参 考 文 献

- 1) Hayes-Roth, F., Waterman, D. A. and Lenat, D. B.: An Overview of Expert Systems, in Hayes-Roth, F., Waterman, D. A. and Lenat, D. B. (eds.), *Building Expert Systems*, pp. 3-29, Addison-Wesley, Massachusetts (1983).
- 2) McDermott, J.: R1: A Rule-Based Configurer of Computer Systems, *Artif. Intell.*, Vol. 19, No. 1, pp. 39-88 (1982).
- 3) Buchanan, B. G., Barstow, D., Bechtal, R., Bennett, J., Clancey, W., Kulikowsky, C., Mitchell, T. and Waterman, D. A.: Constructing an Expert System, in Hayes-Roth, F., Waterman, D. A. and Lenat, D. B. (eds.), *Building Expert Systems*, pp. 127-167, Addison-Wesley, Massachusetts (1983).
- 4) Forsythe, D. E. and Buchanan, B. G.: Knowledge Acquisition for Expert Systems: Some Pitfalls and Suggestions, *IEEE Trans. SMC*, Vol. SMC-19, No. 3, pp. 435-442 (1989).
- 5) Boose, J. H.: Personal Construct Theory and the Transfer of Human Expertise, *Proc. AAAI*, pp. 27-33 (1984).
- 6) Kahn, G., Nowlan, S. and McDermott, J.: Strategies for Knowledge Acquisition, *IEEE Trans. PAMI*, Vol. PAMI-7, No. 5, pp. 511-522 (1985).
- 7) Davis, R.: Interactive Transfer of Expertise: Acquisition of New Inference Rules, *Proc. IJCAI*, pp. 321-328 (1977).
- 8) Suwa, M., Scott, A. C. and Shortliffe, E. H.: An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System, *AI Magazine*, Vol. 3, No. 4, pp. 16-21 (1982).
- 9) Hammond, K. J.: Planning and Goal Interaction: The Use of Past Solutions in Present Situations, *Proc. AAAI*, pp. 148-151 (1983).
- 10) Hammond, K. J.: CHEF: A Model of Case-Based Planning, *Proc. AAAI*, pp. 267-271 (1986).
- 11) Hammond, K. J.: Learning to Anticipate and Avoid Planning Problems through the Explanation of Failures, *Proc. AAAI*, pp. 556-560 (1986).
- 12) Kolodner, J. L., Simpson, R. L. and Sycara, K.: A Process Model of Case-Based Reasoning

- in Problem Solving, *Proc. IJCAI*, pp. 284-290 (1985).
- 13) Kolodner, J.L.: Extending Problem Solver Capabilities through Case-Based Inference, *Proc. Case-Based Reasoning Workshop*, pp. 21-30 (1988).
- 14) Kolodner, J.L. and Simpson, R.L.: The MEDIATOR: Analysis of an Early Case-Based Problem Solver, *Cognitive Science*, Vol. 13, No. 4, pp. 507-549 (1989).
- 15) 奥田健三, 山崎勝弘: 事例ベース形推論とその応用例, *情報処理*, Vol. 31, No. 2, pp. 244-254 (1990).
- 16) 渡辺俊典, 佐久木浩二, 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: 計算機室レイアウト用エキスパート・システムの開発, *情報処理学会論文誌*, Vol. 26, No. 5, pp. 926-935 (1985).
- 17) Wilensky, R.: *Planning and Understanding*, Addison-Wesley, Massachusetts (1983).
- 18) 吉浦 裕, 橋本和広, 辻 洋: Case-Based Reasoning によるエキスパートシステムの知識獲得の容易化(1)—事例利用方式—, 第40回情報処理学会全国大会論文集, pp. 274-275 (1990).
- 19) 橋本和広, 吉浦 裕, 辻 洋: Case-Based Reasoning によるエキスパートシステムの知識獲得の容易化(2)—計算機室レイアウト問題への適用—, 第40回情報処理学会全国大会論文集, pp. 276-277 (1990).

付 録

下記(a), (b), (c)の積が成り立つための十分条件が, (a), (b1), (c1)の積であることを証明する。

(a) AP は, 現在の問題と事例の問題の共通部分問題 Pc の, 事例の問題の解決過程における解である。

(b) AP は, 現在の問題における部分問題間の干渉を補正しなくても, 現在の問題の解決過程における Pc の解として利用できる。

(c) AP は, 事例の問題における部分問題間の干渉を補正しなくても, 現在の問題の解決過程における Pc の解として利用できる。

(a) 上記と同じ。

(b1) 現在の問題のうち Pc 以外の部分問題が, Pc に干渉可能でない。

(c1) 事例の問題のうち Pc 以外の部分問題が, Pc に干渉可能でない。

まず, (b)∧(c)が成り立つための十分条件として, 次の(d)が設定できる。

(d) AP は, 現在の問題の解決過程における Pc の解である。

∴) (d)が満たされれば, AP は, すでに, 現在の問題の解決過程における Pc の解になっているので, 干渉を補正しなくても, そのまま利用できる。したがって, (b)∧(c)が満たされる。

次に, (a)∧(d)を満たすための十分条件は, 以下の(a), (b'), (c')の積である。

(a) 上記と同じ。

(b') 現在の問題において, Pc 以外の部分問題が, Pc に干渉しない。

(c') 事例の問題において, Pc 以外の部分問題が, Pc に干渉しない。

∴) (a)より, AP は, 事例の問題の解決過程における Pc の解である。3.2節(1)で述べた部分問題間の干渉の定義から, (c')より, 事例の問題から Pc 以外の部分問題を除いても, Pc の解は変わらない。したがって, 事例の問題の解決過程における Pc の解は, Pc を単独で解いた場合の解に等しい。逆に, (b')より, Pc を単独で解いた場合の解は, 現在の問題の解決過程における Pc の解に等しい。したがって, (a)∧(b')∧(c')が成立するならば, AP は, 現在の問題の解決過程における Pc の解であり, (a)∧(d)が成立する*。

最後に, (a)∧(b')∧(c')を満たすための十分条件は, 上記(a), (b1), (c1)の積である。

∴) 3.2節(1)で述べた部分問題間の干渉と干渉可能性の定義から, ある問題 P1 が他の問題 P2 に干渉可能でないならば, P1 が P2 に干渉することはない。したがって, (a)∧(b1)∧(c1)が成り立つならば, (a)∧(b')∧(c')が成り立つ。

(平成2年8月30日受付)

(平成3年2月12日採録)



吉浦 裕 (正会員)

1957年生。1981年東京大学理学部情報科学科卒業。同年(株)日立製作所入社。同社日立研究所を経て, 現在, 同社システム開発研究所に勤務。自然言語処理, CAD, 知識処理の研究に従事。1990年本学会全国大会学術奨励賞受賞。電子情報通信学会会員。

*なお, (a)∧(b')∧(c')は, (a)∧(d)の必要条件ではない。例えば, 事例の問題と現在の問題が等しい場合には, Pc として事例のどのような部分問題を探っても, (a)∧(d)を満たす。すなわち, (a)∧(b')∧(c')が満たされなくても, (a)∧(d)は満たされる。