

## 問題解決戦略データベースの一提案とその応用

飯田 栄治

金沢学院大学 美術文化学部 情報デザイン学科

今回、これまで提案した人間の解決戦略を利用した問題解決システムに対し、解決戦略データ管理機能の付加を行った。本問題解決の特徴は、状態空間、オペレータ、ゴール状態、評価関数を問題解決データの基本構成とするものであり、対象問題を複数の部分問題に細分化しそれらの状態空間同士を一定の条件にて連結し、扱い易い問題に再構成することである。そのため解決戦略データベースを作成に関しては考慮すべき制約も多い。本報告では、それらの制約を満足する問題解決戦略知識管理システムを本体システムに組み込んだ際の実装概要を報告するとともに、具体的な問題に応用した結果及び問題点についても述べる。

### A Proposal of Problem-solving Strategy Database and Its Application

Eiji IIDA,

Dept. of Information Design, Faculty of Fine Arts and Informatics, Kanazawa Gakuin University

This time, the author has add database management functions for problem-solving strategy to previously proposed problem-solving system that has utilized human strategy. A basic data structure of this problem-solving system consists of state space, operators, goal state and state evaluation function. In order to construct the system, we divide a target problem to some sub-problems and restructure to an easy problem by connecting each sub-problem under some conditions. Therefore, many restrictions have to be considered to construct the solving strategy database. This paper presents the overview of the database system and refers to the results of applying for tow examples, the issues of this system etc.

#### 1. はじめに

問題解決システムは、古くからシャノンのチェスマシンの論文[1]にさかのぼる。そもそも問題解決とは、ゴールと利用可能な手段が明確に定義されている問題に対し、与えられた状態を探索または何らかの推論を用いながらゴールに導く(または、ゴール状態に変換すること)である。この考え方は、様々な形で発展し、病気の診断を行うエキスパートシステムやロボットの思考、推論を行うシステム、コンピュータチェスマシンなど、各方面で大きな成果を挙げている。しかしながら、更に、広大な情報空間を持つ問題は、人間のヒューリスティックスを多く必要とする。また、特定の職種の洗練された状況判断、体系化された組織活動などは、膨大な探索計算などを使用せず人間の経験的な知識をつなぎ合わせていくことで比較的効率良く問題を解決できること分っている[2][3][4]。ここでは、

人間の持つ問題解決戦略(ヒューリスティックスや経験的知識)を利用することに重点を置いた、問題解決システムに焦点を絞り、そのためのデータベースシステムの試作について報告する。以降、2章では対象問題を定義し、3章では、問題解決システム、4章では、データベースシステムへの機能付加、5章では、例題による実験結果を示す。6章で結論を述べる。

#### 2. 対象問題

「問題解決」について議論する場合、対象とする問題のクラスがどのようなものを定義する必要がある。しかし、問題クラスの分割や体系の考え方は多様であり、一概に定義することは困難であるが、ここでは、一般に、エキスパートシステムの議論で行なわれる問題タイプの分け方を基に対象問題の特徴を述べる。

ここで議論する問題は、図1に示すように、「多様なパラメータからなる問題空間であるが、人間にとってある程度問題解決の方向性が分かっている」問題を中心に議論する。対象問題は問題解決のストーリーが分かっていることを前提にしているので、一見システム構築は容易と思えるが、実際には巨大な状態空間が対象でありその解決プロセスは無数にある。

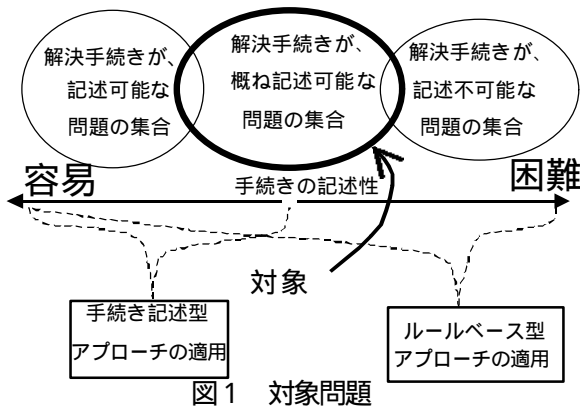


図1 対象問題

### 3 問題解決システム

#### 3.1 提案する問題解決方式の枠組み

既に、[2][5]の中で、人間の持つ問題解決戦略を効果的に利用するための問題解決方式を提案した。一般に、人工知能における問題解決の枠組では、 $S$ を問題空間、 $W$ をゴール空間、そして、 $R$ をオペレータ集合とすると、問題  $P = \langle S, W, R \rangle$  の3項組にて表現できる[6]。本提案方式の枠組みでは、評価関数  $H$  を加え、

$$P = \langle S, W, R, H \rangle \quad (1)$$

にて定義する。既に述べたように、本方式はある問題において、その専門家が持っている問題解決ストーリーを部分問題の系列として抽出しそれを計算機上に知識表現することで問題解決を行なう方式である。図2に示すように、 $K+1$ 個の部分問題にて構成される問題解決ストーリーがあるものとし、任意の部分問題を

$$P_i = \langle S_i, W_i, R_i, H_i \rangle \quad (2)$$

とする。このとき、以下の条件、

- (1) 各部分問題間には解決順序が存在し、終端にゴールが存在すること。
- (2) 隣接する部分問題は一方のサブゴールを共有することで空間結合されていること。
- (3) 各部分問題毎、それに適したオペレータが

割り当てられているものとする。

- (4) オペレータを選択するための評価関数は、単調増加または単調減少関数であること。

を前提として定式化を行なう。

$W_0$ は、問題全体をとおしてのゴールである。なお、全部分問題は以下の条件を満足していること。任意の問題  $P_i$  において

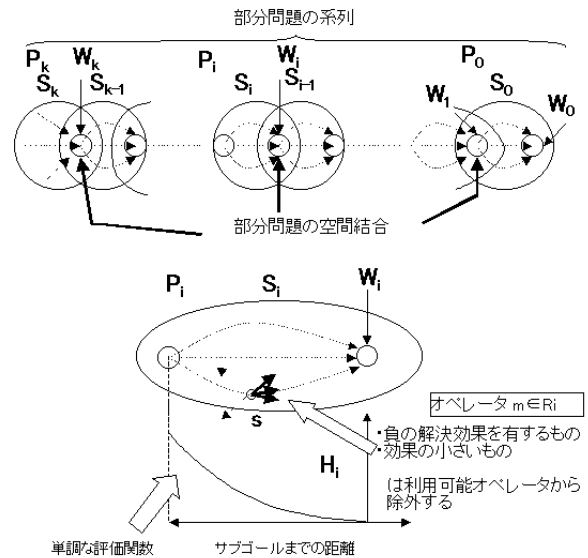


図2 問題解決モデル

$$(1 \quad i \quad k) (S_{i-1}) (S_i) (W_i \quad S_i \& W_i \quad S_{i-1}) \quad (3)$$

ここで、さらに各  $P_i$  をグラフの節点、または、隣接する部分問題同士  $(P_i, P_{i-1})$  をグラフのエッジと考えると、図2は、図3のように単純な有効グラフと考えることもできる。このとき、節点の集を  $V$ 、エッジの集合を  $E$  として、以下の様に記述できる。

$$\langle V, E \rangle = \langle \{P_k, \dots, P_0\}, \{ (P_k, P_{k-1}), \dots, (P_1, P_0) \} \rangle \quad (4)$$

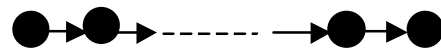


図3 部分問題のグラフ

このグラフは、問題解決の流れを表したものであり、各部分問題のインデックスは、問題解決のフェーズを表しており、フェーズフローと呼ぶこととする。一般に問題解決の流れは複数のフェーズフローを用いて表現すべき問題の方が遙かに多い。そこで、図4のように、複数のフェーズフロー  $F_0, \dots, F_N$  が有機的に結びついたものを考えフェーズ木ということとする。

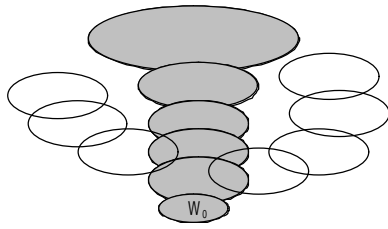


図4 フェーズ木

計算機上で処理する際には、フェーズ木のままでは扱い難いので、フェーズ木を図5の様に展開したものを考える。処理上、最多の部分問題からなるフェーズフローを基準に各フローの部分問題数(j)を決定する。

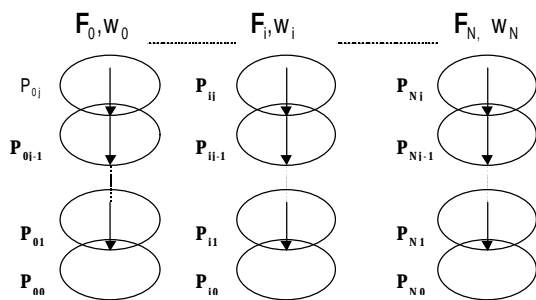


図5 フェーズ木の展開

ここで、状態  $s$  がどの部分問題に属しているのかということを判定する為の条件を  $C_{sij}$  ( $i$ : フェーズフロー番号、 $j$ : フェーズ番号) とすると、以下の手順を順次進めることで問題解決が可能となる。

(1) 問題発見

すべての部分問題空間の条件  $C_{sij}$  について問題の状態が、その条件に合致するか否かを調べる。

(2) 問題選択

(1)にて条件を満たしたフェーズフローに対し、以下の優先順位を以って問題を解決する。

- a. 現在の状態  $s$  が、あるフェーズフローの2箇所以上の部分問題に含まれるとき、フェーズ番号の小さい方を優先。
- b. 現在の状態  $s$  が、2本以上のフェーズフロー上の部分問題に含まれるとき、フェーズフローウェートの大きい方を優先。

(3) 問題解決

(2)にて選択された部分問題の問題解決を行なう。問題解決は、その部分問題において利用可能な全て

のオペレータを仮想的に適用し生成した各状態を評価関数を用いて評価し、最良の評価が得られるものを決定すること問題解決が1ステップ前進する。

3.2 提案する問題解決システム

提案する問題解決システムは、図3のように問題発見、問題選択 及び 問題解決の3つ過程を繰り返すことで最終ゴールへの到達を目指す。

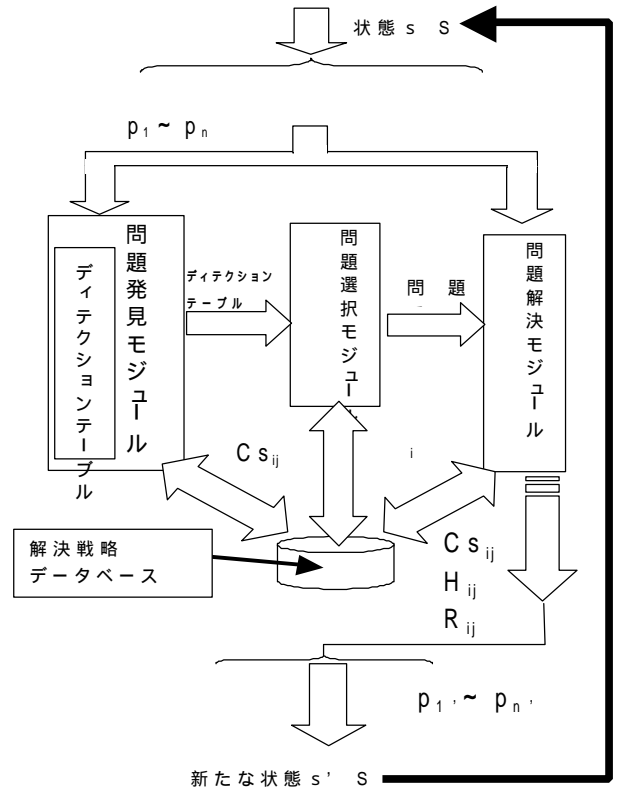


図6 システム構成

解決戦略データは、フェーズフローのノード構成データであり内容は表1のとおりである。

表 1 解決戦略データ内容

項目	定式上の記号	実装上の扱い	実装上の記号
問題Pの問題空間	$S$	条件式	$Cs$
問題Pのゴール空間	$W$	条件式	$Cw$
問題Pのオペレータ	$R$	状態遷移関数の集合	$R$
問題Pの評価関数	$H$	評価関数	$H$
部分問題 $P_{ij}$ の問題空間	$S_{ij}$	条件式	$Cs_{ij}$
部分問題 $P_{ij}$ のサブゴール空間	$W_{ij}$	条件式	$Cw_{ij}$
部分問題 $P_{ij}$ オペレータ	$R_{ij}$	状態遷移関数	$R_{ij}$
部分問題 $P_{ij}$ オペレータ	$H_{ij}$	評価関数	$H_{ij}$
フェーズフローウェート		定数	$i$

#### 4. データベース化の取り組み

既提案システムの問題解決戦略データベースは表 1 の各種データファイルとしてプログラムの中に組み込まれ、データの追加、修正、削除に関しては、ファイル自体に手を加え、再コンパイルする必要があった。そこで、プログラムから完全に独立させデータの入力、変更等が容易に行えるシステムの構築を試みた。本問題解決戦略データベースを構築する際注意すべきことは、既に述べた様に問題解決順序に重要な意味を持つ。マンマシンインターフェースを考える上では、特に、以下の事項に注意する必要がある。

- ・ 隣接するノードにおいては、上流側のノードのサブゴールは、下流側のノードに含まれるなければならない。
- ・ あるノードのデータを変更するという事に際し、データ整合上のミスを防止するために、該当するノードの構成データを同時にすべて表示されるべきである。

以上の要件を考慮し図7に示すようなデータ入力インターフェースを試作し、各ノードを構成するデータを一括して確認できる様にした。また、ノード構成データの新規入力完了すると隣接ノードとの接合チェックが自動的に行われることとした。

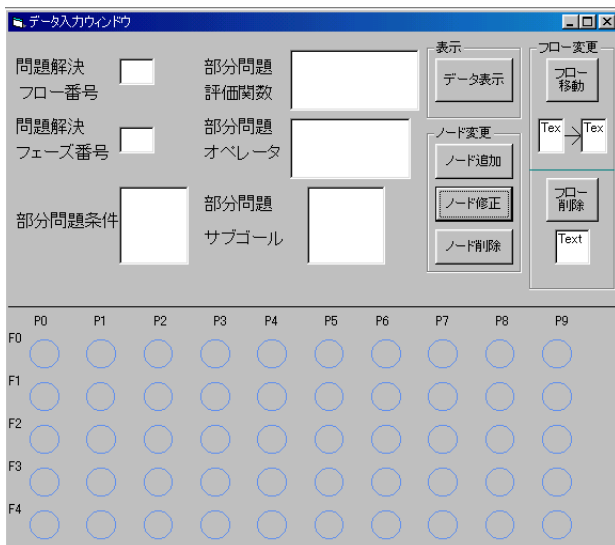


図7 データ入力ウィンドウ

また、図8のようなシステムチェック画面を立ち上げることにより、指定ノードに対しそのノードの定義域に該当する状態を指定またはランダムに生成することにより正常に次ノードに移行可能か否かできる。

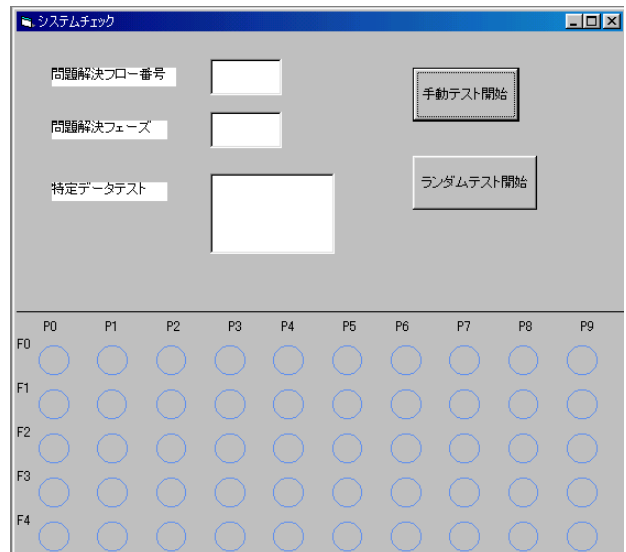


図8 システムチェックウィンドウ

現在チェック中のノードについては、ノードの表示が反転して実行状況が分る。また、エラーがあった場合には、その内容がメッセージとして表示される。

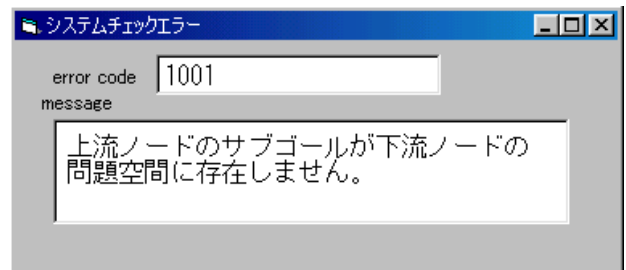


図8 エラーメッセージ

#### 5. 応用例

本章では過去に実施した 15 パズルの例題をこのシステムの枠組にて再構築した事例と仮想空間で活動する組織エージェントの制御の事例について述べる。

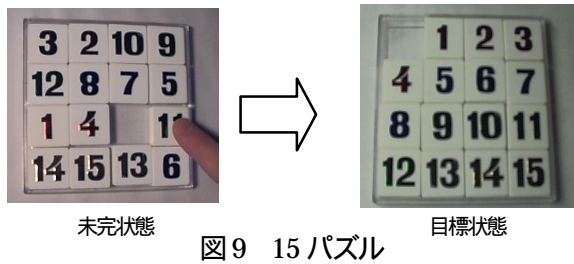
##### 5.1 15 パズル

###### 5.1.1 本例題の概要

一般に、 $N^2 - 1$  パズルは、複数のタイル配置を一箇所のワーキングスペース(ブランク)を利用しながら配置を少しずつ変更し目標状態に並び換えるパズルである。



特に、図9の様な15パズルは玩具店などで市販されている。ここでは、この15パズルを用いる。



### 5.1.2 解決戦略の概要

15パズルは、図10の様な解決プロセスをたどり、解決が可能である。実際に、15名の被験者を募り確認した結果、比較的多くの被験者が、下段、または、上段から順にタイルを揃えていった。また、揃った段は以降極力崩さないよう全体を完成させることが分った。そこでこの戦略を、「人間が直感的に思いつく問題解決戦略をデータ」と位置付けて利用することとした。

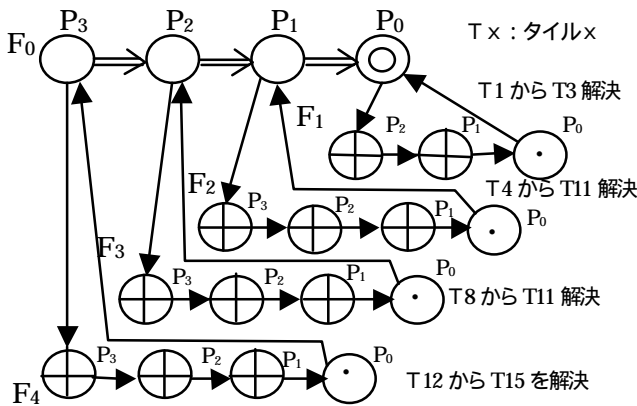


図10 15パズルの解決過程

図10では、問題解決過程において各段毎の解決を本流（主フロー）と各段の個々の解決を支流（副フロー）として図解している。このフェーズフロー図解用記号の意味は以下のとおり。

- : 問題解決の主フローのサブゴール
  - : 問題解決の主フローのゴール
  - + : 問題解決の副フローのサブゴール
  - : 問題解決の副フローのゴール
  - ⇒ : 主フローの部分問題リンク
  - : 副フローの部分問題リンク
- (以降も同様)

また、分析の結果、人間が主に使用する基本操作は、以下のとおりであった。それぞれ OP1~OP3 として問題解決の中で利用するものとする。

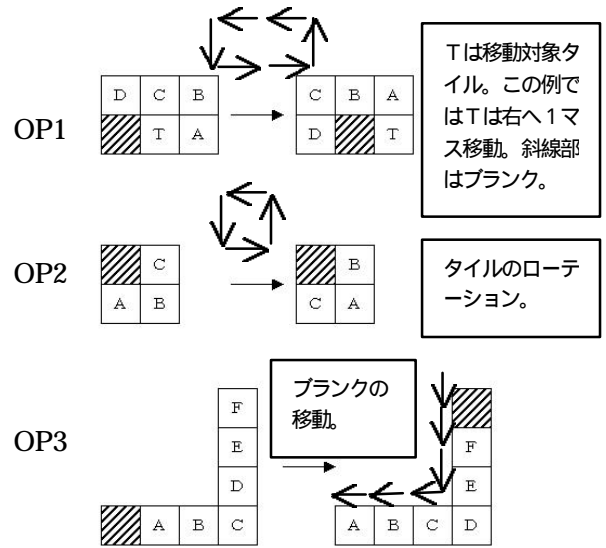


図11 オペレータ

### 5.1.3 データベースへの入力

各ノードデータの入力に関しては、パズルの視覚イメージで入力できるようにした。例えば、図12は、ノード(F3, P2)のデータ入力例を示す。ただし、部分問題条件とサブゴールの\*マークは、任意のタイルまたはブランクを意味する。また、便宜上、10から15の数字は a,b,c,d,e,f として表現している。



図12 入力例

ここで、ノード(F3,P2)のサブゴールは、ノード(F3,P1)の部分問題条件に含まれる(一致する)ようにデータを入力してデータを構成していく。これにより、これらのノードはリンクされていることとなる。集合的に見れば、ノードのリンクは、図13のように、ノード(F3,P2)に対して自己のサブゴールが内包されることで接続されている」のが元々の考え方である。

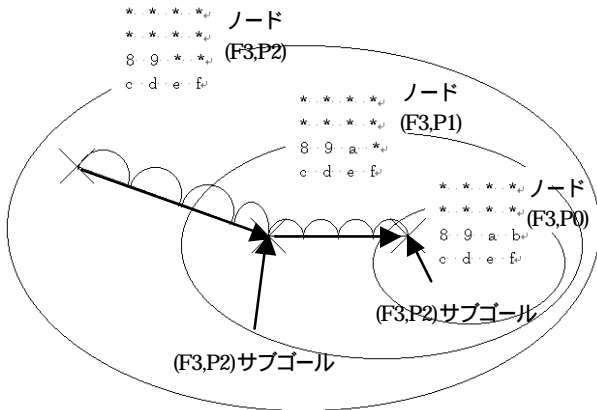


図 13 ノードの接続

### 5.3 問題解決結果

本データによる問題解決の結果の実施結果は、報告 [4] したとおりである。図 14 において、及び は提案方式の実験結果であり、深い探索を用いずとも高速にかつ良質な解を得ることが出来た。RTA\*[7]は深さ 5,8 および 10 による実時間探索結果である。

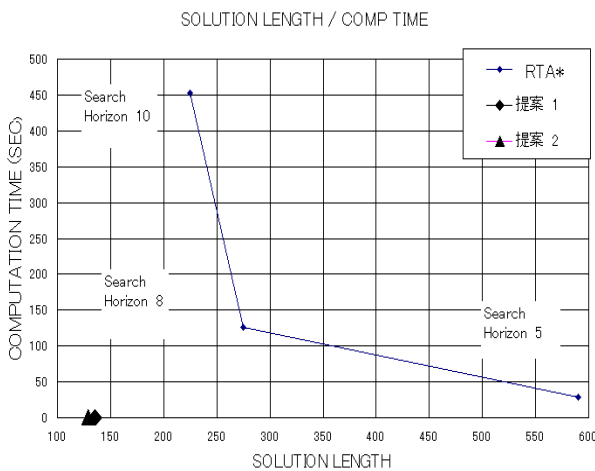


図 14 問題解決結果

### 5.2 組織的マルチエージェント

ここでは、組織的マルチエージェントの問題解決の問題に本システムを応用する。近年、仮想環境において複数の多様なエージェントの自律的な行動を記述する方法は各方面から注目されている。例えば、ゲーム業界においては、ゲームの臨場感を盛り上げる意味で複数の人物や移動するオブジェクトを登場させることは常識になりつつある [8]。しかしながら、エミュレー

ションによるものではなく個々が自律的に行動するエージェントである場合、協調計算手法、計算負荷の低減など多くの解決すべき問題があり、様々なプロジェクト、例えば、“ロボカップレスキュー [9]” 等で新しい挑戦が行われている。

#### 5.2.1 本例題の概要

本例題は一種のウォーゲームであり利害関係のある 2 つのグループが攻防を広げると言う意味では先の例と類似している。その内容は、16 両の車両に見立てたエージェント (中隊規模) 同士が戦略拠点を巡って攻防を行うものである。図 15 は例題イメージであり攻撃側 (Blue) と防御側 (Red) が存在している。

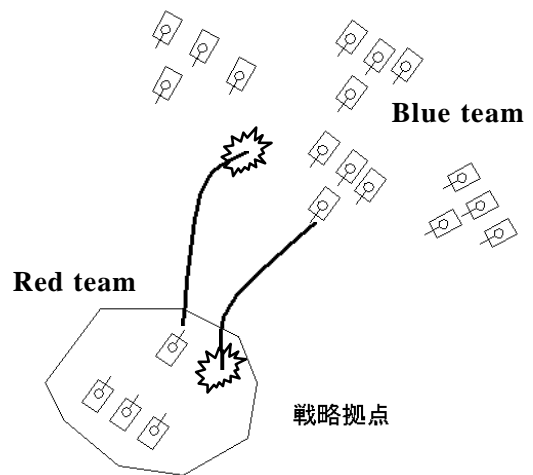


図 15 例題イメージ

米国では、CGF (Computer Generated Force) と呼ばれる模擬部隊発生装置が既に開発されており、仮想的に大部隊を発生させ、これを用いて指揮訓練をするようなシステムも存在している。この応用として、有事危機管理のシミュレーション等も可能であろう。

#### 5.2.2 解決戦略の概要

一組織は 4 つのサブグループ (小隊) からなりサブリーダーが存在する。その内の一つのサブリーダーが組織全体 (中隊) のリーダーも兼務する。また、サブグループは 4 車両から構成される。図 15 のとおりである。この組織構造により、戦略データは、リーダー、サブリーダーおよび他の構成メンバの各階層毎に問題解決戦略を構築する。

リーダー用には、組織をまとめて戦略拠点攻略の目的を達成するための知識を与え、下位の構成メンバには上位階層の発する命令に対処するための知識を与える。

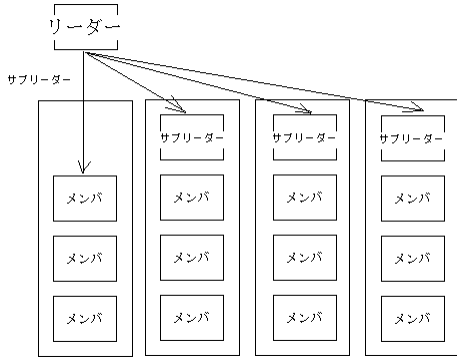


図16 組織構造

提案システムは一見複雑に見える以下の状況にも対処可能な柔軟性を持っている。

(1) 命令の伝播

各エージェントの行動は、ノーマル、攻撃、撤収などからなり、ノーマル時は戦略拠点への進攻であり、構成メンバーはリーダーの後ろを一定の隊形を構成して追従する。途中、敵と遭遇した時は、先に示したとおりリーダーの命令により攻撃行動を行う。図17は攻撃命令が順次下位階層に伝播している様子である。

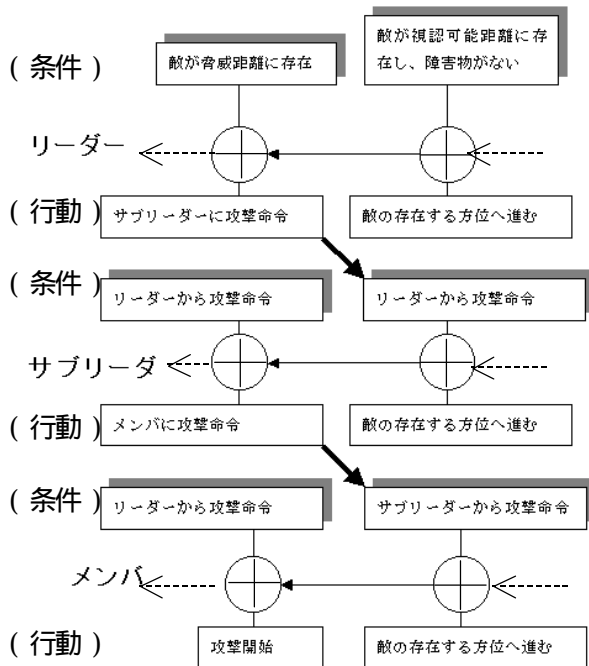


図17 階層における情報の伝達

万が一、リーダーがダメージを受けた場合は撤収する。

(2) 解決戦略のオーバーライド

下位の構成エージェントが局所的に発生するイベントの対応するために上位エージェントの命令をオーバーライドすべき状況が存在する。例えば、隊形を組みながらの進攻中または攻撃の最中の障害物の回避などである。このような場合、フェーズフロー番号の大きい方を優先戦略としており図18の様に回避可能となる。

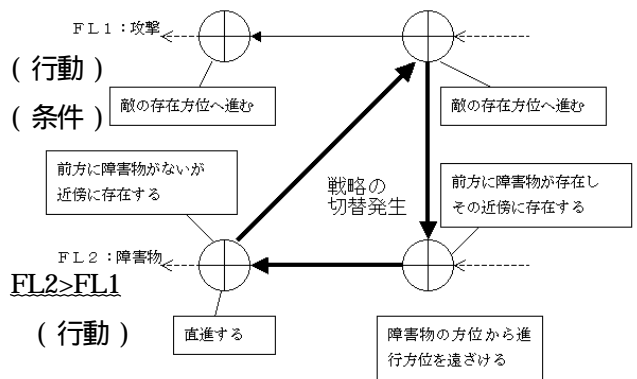
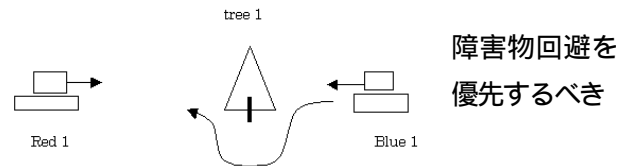


図18 戦略の切替

5.2.3 データベースへの入力

本例題のデータ入力例は図19のとおりである。

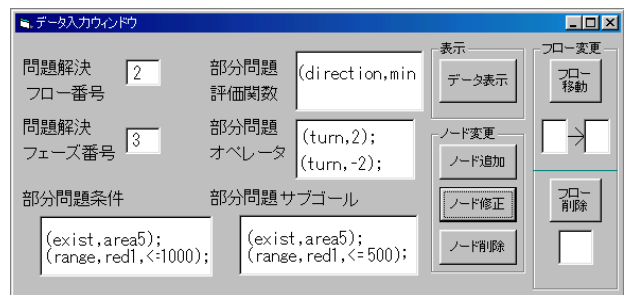


図19 入力データ

「部分問題条件」は、条件を列記していくことで定義する。例えば、「エージェントは、エリア5に存在し、Red1に対して距離1000m以下」という条件は

```
(exist,area5);
(range,red1,<=1000);
```

となる。この条件を判定しフェーズを認識する。次に、本フェーズ下で使用するオペレータ turn, forward, back, stop(旋回,前進,後退,停止)とレベルを入力する。

例 (turn, -2), (forward,2), (back,1), (stop); 等

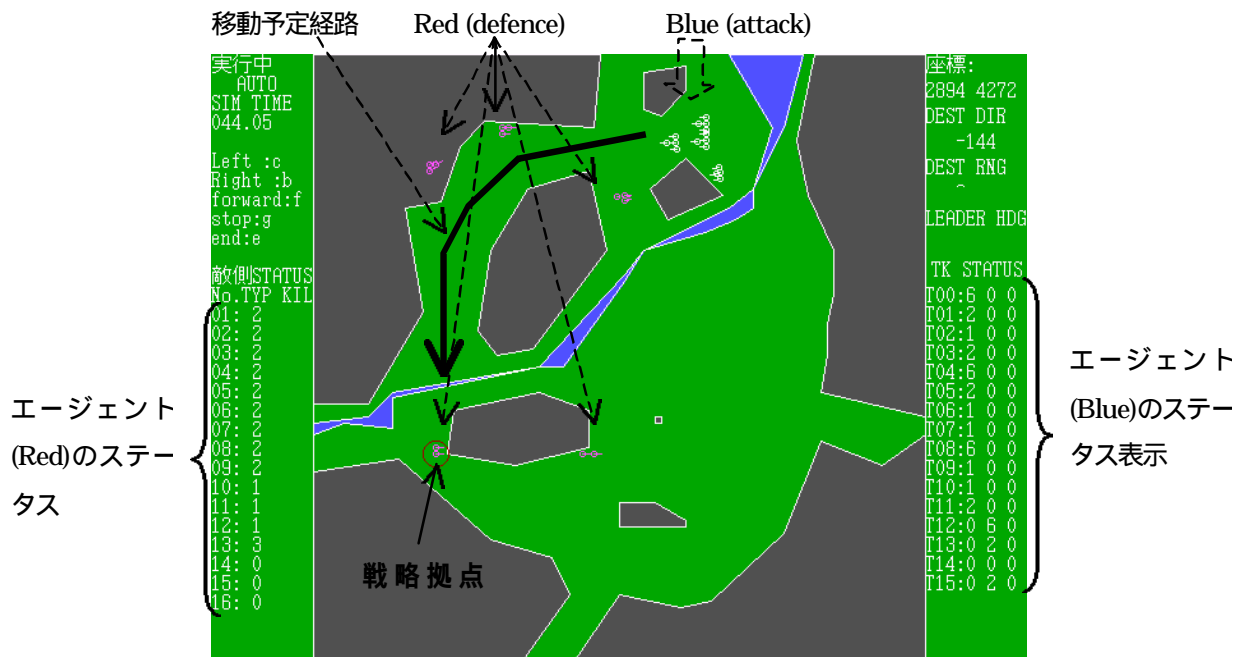


図 20 シミュレーションの様子

部分問題評価関数は、相対距離や相対方位の最小、最大などが主に用いられ、これらは結果的に単調減少の評価関数となる。例えば、

(range, max, red1); 「red1 への距離が最大」  
(direction, min, red1); 「red2 への方位が最小」

これらにてサブゴールに導くオペレータを選択する。

### 5.2.3 問題解決結果

戦略や攻撃能力に関し、攻撃側 (Blue) が有利になるように調整した上で、図 20 のようなマップ上でゲームの状況変化を確認したところ、途中、特に違和感もなくシナリオが進み、Blue が戦略拠点攻略に至った。

## 6. おわりに

今回は、同一のシステムの枠組みを用い、全くタイプの異なる問題を、インターフェースなどの多少の変更のみで解決することができた。一般に、問題の規模や性質によっても異なるが、今回取り上げた例題では、一人の設計者が2週間程度の期間で構築可能であった。特に、後者の例題に関して、システム完成後、3D CG と連結して確認したところ大きな問題もなく、想定したシナリオが実現できており、生産性と完成システムの安定性の点で満足できるものであった。

一方、問題点として、処理の都合上部分問題の数が固定されることやデータベース作成のユーザーインタ

ーフェースとしてマウスだけでノードのリンク構造を変更できないなど改善すべき点も残されている。

## 謝 辞

「組織的マルチエージェント」の例題は三菱プレジジョン(株)との共同研究の中で発案されたものであり、3D CG への接続および内容についての議論をして頂いた中山利徳氏、伊藤宏明氏に感謝する。

## 参考文献

- [1] Shannon,C.C., "Programming a computer for playing chess",Philosophical Magazin (Series 7), pp.125-133, 1950.
- [2] 飯田栄治,下平博, "実時間意思決定システムの一提案とその応用", 人工知能学会, 第12回全国大会,講演論文集, pp.528-529, Jun., 1998
- [3] 飯田栄治,下平博, "航空機パイロットの知識構造に関する一考察", 日本認知科学会,第15回全国大会,講演論文集, pp.68-69,Jun.1998
- [4] 飯田栄治, 國藤進, 下平博, 木村正行, "N<sup>2</sup>-1 パズルのスケールダウン解法", 電子情報通信学会, 和文論文誌 D-I,pp.504-514, Jun.,1998
- [5] 飯田栄治, 下平博, "航空機パイロットの知識構造に関する一考察", 日本認知科学会, 第15回全国大会, 講演論文集, pp.68-69,Jun.1998
- [6] R.B.Banerji, "人工知能 - コンピュータによるゲーム", 共立出版, 高原他訳
- [7] R.E. Korf, "Depth-First Iterative-deeping An Admissible Tree Seach", Artificial Intelligence, Vol.27, No.1,pp.97-109, 1985
- [8] 斎藤直宏, "PS2 でゲームはこう変わる!", 情報処理学会, Vol.41, No.6, June 2000
- [9] 田所諭 他, "ロボカップレスキュープロジェクト", 人工知能学会誌, 15 巻 5 号, 2000, 5