

TM-0793

代替案選択支援機構における
主観的評価の一貫性保持

新谷虎松(富士通)

September, 1989

©1989, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

代替案選択支援機構における主観的評価の一貫性保持 A Consistency Maintenance Mechanism for Subjective Judgments

新谷 虎松

Toramatsu Shintani

富士通(株)国際情報社会科学研究所

IIAS-SIS FUJITSU LIMITED

tora@iias.fujitsu.junet

代替案の選択を効果的に支援するための機構について論じる。本機構は、意思決定に必要な基本的評価基準(考慮すべき要素)の抽出・整理を支援し、ユーザの主観的評価(価値判断)を積極的に蓄積・利用する。主観的評価は、AHPに基づく一対比較法を用いて重要度として定量化され、代替案を評価・選択するための情報として用いられる。一対比較による主観的評価の整合性を自動的に保存するために、Prolog上に構築されたプロダクションシステムKOREAEの非単調推論機能に基づくTMS機能を用いた、本機能により、主観的評価の一貫性は効果的に検査・保持される。

1. はじめに

一般に、知識ベースに基づく問題解決支援システム(例えば、エキスパートシステム)において、知識ベースは前もって構築され、繰り返し用いられ、そして洗練化されることが前提となっている。そのため、知識ベースの構築・洗練化には多くの時間が必要とされる。一方、問題解決支援システムにおいて、このような知識ベースを前もって構築すること自体が困難もしくはは無意味な対象領域が多く存在している。例えば、知識ベースを繰り返し用いない(頻りに参照しない)ならば、前もって知識ベース構築に多くの時間をかけることは無意味なことである。本論文では、このような知識ベースの構築が困難な分野における問題解決支援を対象としている。具体的には、ユーザの知識を積極的に抽出し構造化することを支援することにより、問題解決を支援する機構について論じる。本機構はプロダクションシステムKOREAE[新谷 87, Shintani 88]を用いて実現され、代替案選択支援機構CDSS(Choice Design Support System)[新谷 88b]と呼ばれる。

具体的に扱う問題は、代替案の中から最適な案をひとつ選ぶ問題を取りあげる。この問題は、多目的意思決定[中山 86]を代表する問題であり、代替案を選択する過程でいくつかの競合する要求を同時に成り立たせようとするために生じる。CDSSでは、ユーザの知識を積極的に抽出し構造化することを支援するために、ユーザの主観的評価(価値判断)を積極的に利用・蓄積する。これにより競合する要求の重み付けを明らかにし、代替案の選択を効果的に支援する。具体的には、ユーザの主観的評価は、AHP(Analytic Hierarchy Process)[Saaty 80]に基づく一対比較法を用いて重要度として定量化され、代替案選択における競合を解消するための情報として用いられる。本論文では、特に、CDSSにおける支援機能を実現する上で最も重要と思

える主観的評価の一貫性保持機構に焦点を当てて、その実現方式について詳述する。

2. 代替案選択支援機構CDSS

CDSSにおける支援過程は、図1のように示すことができる。図1で示すように、CDSSはユーザの主観的評価を積極的に利用することにより、解決すべき問題の目標及び副目標の設定を支援し、問題の構造を明らかにするための機能を提供する。具体的には、考慮すべき要素を階層的に整理し、要素の重要度を明らかにすることを通してユーザ知識の構造化を支援するとともに問題の解決を支援する。このように構造化された知識は、将来、知識獲得支援システムの基本的な入力情報として用いられることを想定している。

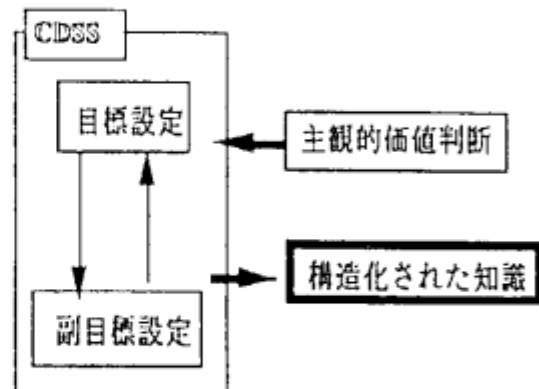


図1 ユーザ知識を用いた問題解決支援過程

CDSSは図2で示すように(1)階層設計支援機構、(2)主観的評価支援機構、および(3)アウトライン・プロセス機構により構成される。CDSSにおけるユーザ知識の利用による問題解決はこれら(1)~(3)の機構を組み合

わせて用いることにより達成される。(1)の階層設計支援機構は、意思決定問題において考慮すべき要素がどのようにして全体の問題を構成しているかを階層的及び視覚的に整理し明らかにすることを支援する。この階層化により、問題における複雑さが解消され、考慮すべき要素間の関係が抽出される。本機構は、ネットワーク管理システムKORE/EDEN [新谷 86a] 及び、グラフ描画・編集システムKORE/GET(GraphEditing Tool)をベースにして構築され、主に、問題を明確化するための視覚的知識構造化支援環境を提供する。(2)の主観的評価支援機構は、主観的評価法のひとつであるAHP [Saaty 80]を用いて一対比較による主観的評価の定量的取扱いを支援し、ユーザの主観的評価を積極的に導入するために用いられる [新谷 88b] (本機構は、本論文の中心課題であり、第3節で詳述する)。(3)のアウトライン・プロセッサは知識の断片を整理し構造化するために用いられる。ここで整理・構造化された知識は(1)の階層設計支援機構でさらに視覚的に明確化される基本的な情報を提供する。

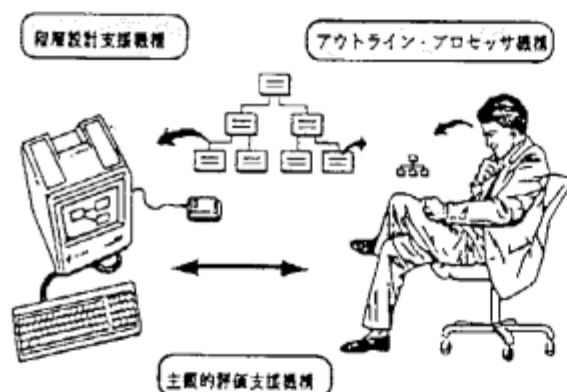


図2. CDSSの構成

図3は階層設計支援機構における階層エディタの表示例を示している。エディタ機能はグラフ描画・編集機構KORE/GETを用いて実現される。編集機能として、(1)グラフのノード及びアークの編集、(2)グラフの整画機能等がある。この例では、プリンターの選定(レベル1)に関して、考慮すべき要素(レベル2)を列挙・明確化する事により、代替案(レベル3)における要素間の優劣を明らかにするものである。

CDSSでは、階層設計支援機構と主観的評価支援機構を協調して用いることにより意思決定におけるシミュレーション機能を実現する。これは、あるレベルの要素間の一対比較による重み付けの値を少し変化させることにより、この変化が結論にどう影響を及ぼすかを見ることである(感度分析を行なうことに相当する)。これにより、考慮すべき評価基準を増やしたり、新たな階層を設計する指針となる。このようなシミュレーションを通してユーザは意思決定問題に関連して考慮されるべき知識(もしくは問題)の構造化を実現する。

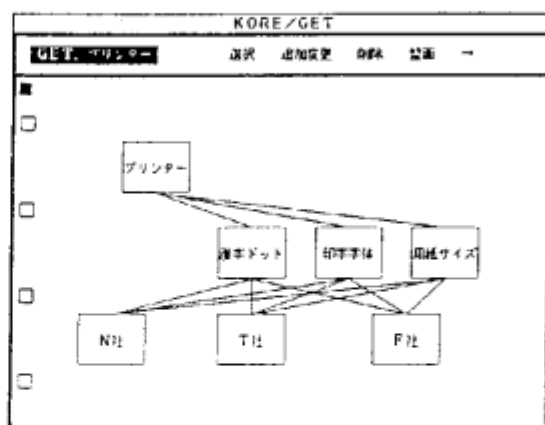


図3. 階層エディタ表示例

3. 主観的評価の一貫性保持

3.1 主観的評価手法

要素間の優劣は主観的評価により重みの大きさで決定される。主観的評価支援機構は、要素間の重み付けを一貫性をもって行なうために用いられる。ここで用いられる主観的評価手法はAHP [Saaty 80]に基づく一対比較法であり、9点法が用いられる。本手法における一対比較は、一つ上のレベルの要素を具体的な評価基準として用いることにより、その重み付けを達成する。例えば、図3においてレベル2の要素間の重み付けは、レベル1の要素(ここでは、「プリンターの選定」)を評価基準として用いることにより、決定される。この例では、レベル2における「漢字ドット数」と「印字字体」(そのほかの組合せとして、「漢字ドット数」と「用紙サイズ」及び「印字字体」と「用紙サイズ」がある)がレベル1の「プリンターの選定」に対して、どちらがどのくらい重要かを9点法による一対比較により主観的に判定する。最終的に、「プリンターの選定」に関するレベル2の各要素の重みは、一対比較で得られた主観的数値(重み)を割り付けることにより決定される。これら重みは重要度と呼ばれる。

具体的には、重要度は一対比較により得られる一対比較行列 [Saaty 80] をもとに固有値や固有ベクトルを求めることにより解析的に得られる。つまり、本手法では、最大固有値に対する固有ベクトルのそれぞれの値が対応する要素の重要度となる。例えば、先の「プリンターの選定」において次のような一対比較行列が得られる。

	漢字ドット数	印字字体	用紙サイズ	
漢字ドット数	1	3	2	1: 同くらい重要 3: 若干重要 5: 重要 7: かなり重要 9: 絶対に重要 2,4,6,8: 能動的に利用
印字字体	1/3	1	1/2	
用紙サイズ	1/2	2	1	

ここで得られた一対比較行列の要素 a_{ij} は項目 I_i の重みと項目 I_j の重みを一対比較して得られた相対的重み付けであり、9点法により1（同じくらい重要を示す）から9（絶対的に重要を示す）までの一対比較値（正整数値）のひとつをとる。この行列の特徴は、(1)対角要素が1、(2)要素の値は $a_{ji}=1/a_{ij}$ である。そこで、 n 個の項目に対して $n(n-1)/2$ 回の一対比較で項目間の重要度を決定できる。具体的には、先に述べたように、ここで得られた行列の最大固有値に対する固有ベクトル $[0.540, 0.163, 0.297]$ が得られ、“漢字ドット数”、“印字字体”、“用紙サイズ”に対してそれぞれ0.540、0.163、0.297の重要度が算出される。

一対比較法において、同一レベルの各要素は互いに独立、もしくは独立に近い関係であることが望ましい。なぜなら、従属的な要素を含む一対比較は判断を誤らせる原因となる。そこで、このような従属要素による不都合を避ける手段として、CDSSでは、従属する要素を一つのグループにまとめレベル分けをする（ひとつ下のレベルにする）ことを支援する。これにより、従属要素は新たなレベルで（つまり、新たな評価基準に従って）独立的に比較することが可能になる。このグループ化は、類似したものどうしをまとめることに相当し、対象とする問題の明確化・構造化を支援するための基本的な情報になる。類似したものどうしかどうかを判定する基準として、評価基準に対する重要度を用いることができる。つまり、考慮する要素において、全ての評価基準に対して求められる重要度に差がなければ、これら要素は類似しているものとしてグループ化できる。CDSSでは、差の許容範囲として10%を用いている。この10%という値は、従属性を測定するための経験的な目安であり、情報の構造化及び従属性の測定に関連して今後さらに検討を要するものである。レベル分け機能は、具体的には、次のように説明できる。例えば、図4左のレベル2における要素(3)と要素(4)の間に従属関係があると判定された場合、CDSSは図4右のようなレベル3の階層が新たに作成される。図4右のレベル2の新たな要素(3)(4)はレベル3の要素(3)と(4)の新たな評価基準となる。

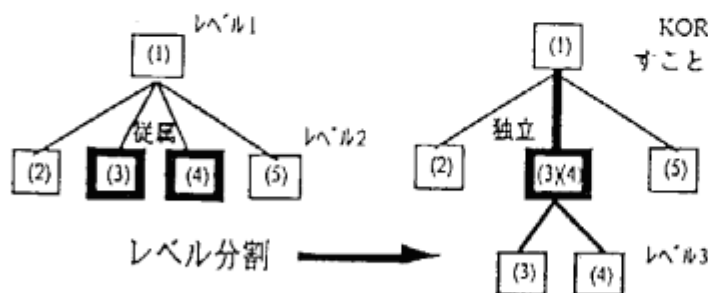


図4. レベル分け機能

さらに、同一レベルの要素数が7より大きくなる場合もレベル分けされる。これは、意思決定者の一度に扱える適切な数（ 7 ± 2 ）より得られた。これらの知識はKORE/IE [新谷 88a] を用いて、ルールとして表現・利用される。

AHPに基づく一対比較による重み付けを用いる利点は、主観的評価の整合度を表す指標があることである。

この指標は非整合度尺度(I.R.(inconsistency ratio))と呼ばれ、次のように求めることができる。

$$I.R. = (\lambda_{\max} - n) / (n - 1) \quad \dots (1)$$

ここで、 $\lambda_{\max}$ は先に述べた一対比較行列より得られる最大固有値であり、 n は行列のサイズ（一対比較の要素数）である。理想的な一対比較が行なわれると非整合度尺度I.R.は0になる。もし、この非整合度尺度の値が0.1以下であれば、経験的に、一対比較に整合性がありと判断される。そして、非整合度尺度の値が0.1より大きくなった場合には、一対比較に整合性がないことを意味しており、一対比較のやり直しが必要となる。一般に、一対比較のやり直しは、煩わしい比較作業の繰り返しが再度強いることになり、意思決定者にとり大きな負担になる。CDSSでは、このような負担を軽減するために、重み付けの整合性を保持するための機構を有している。

3. 2 主観的評価支援機構

主観的評価支援機構は、既に得られた一対比較による重み付けを自動的に ± 2 の範囲で微調整することにより、重み付けの一貫性を維持する。この微調整の範囲は経験的（及び実験的）に得られたものである。これは、一度に比較すべき要素数（最大7つ）を制限したことにより、一対比較法から得られる結果がそれほど矛盾していないことを前提としている。本機構は、プロダクションシステムKORE/IEにおける非単調推論機能を用いることにより効果的に実現している [新谷 88a, Shintani 89]。KORE/IEはProlog上に構築された高速度前向き型プロダクションシステムであり [新谷 87, Shintani 87b, 88]、論理プログラミング及びOPS 5 [Forgy 81] 風のルールプログラミングを提供する。

3. 2. 1. 重み付けの実行

KORE/IEを用いた重みの自動調整は図5のように示すことができる。

図5の枠で囲まれたシンボルはルール名を表している。ルール'start'はCDSSにおいて一対比較を始める際に必要となる初期設定を行なう。ルール'comparison2'は、3.1節で述べた一対比較行列の対称要素（主対角線に関して対称な位置にある要素）を決定するために用いられる。ルール'comparison2'は次のように記述される。

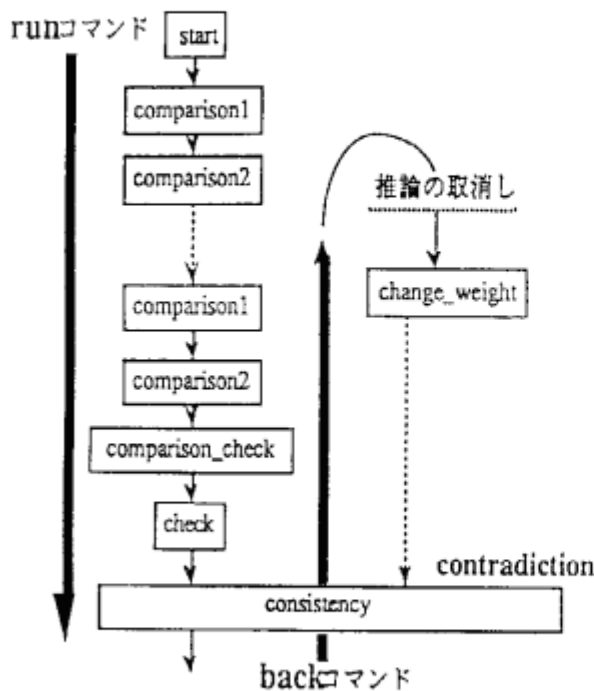


図5. KORE/IEによる重みの自動調整

もし、『一対比較行列XにおけるC行R列の要素の重みがW』であり、かつ、
『一対比較行列XにおけるR行C列の要素の重みが未定』であれば、
その時、
『一対比較行列XにおけるR行C列の要素の重みを $1/W$ 』とする。

具体的には、ルール“comparison2”はKORE/IEを用いて次のように表現される。

```

comparison2:
if matrix_element(name=X,column=C,row=R,
weight=(W > 0)) &
- matrix_element(name=X,column=R,row=C)
then
add(matrix_element(name=X,column=R,row=C,
weight=compute(1/W)))
  
```

ここで、LHS(Left Hand Side)の2つめのシンボル“-”を伴ったパターンは負のLHSパターン（1つめは正のパターン）と呼ばれ、WMにこれとマッチするWM要素がない場合に真となる。また、負のLHSパターンは、WMから負のLHSパターンとマッチするWM要素が削除された際にも真となる。LHSパターンは、例で示すように、Prologの項に相当し、そのファンクター名をクラス名とよび、引数としていくつかの“スロット=値”対であるスロット記述（シンボル“=”の左辺をスロット名、右辺をスロット値と呼ぶ）により構成されている。複数のLHSパターンやRHSアクション（ルールのRHS(right hand side)におけるアクション）はシンボル“&”を用いて区切られる。

ルール“comparison1”は、具体的な一対比較を行なう

ために用いられる。その際、ユーザは入力する一対比較値に対して(1)確実な値、(2)仮の値といった区別を表明する。この情報は、一対比較に整合性がないと判定された場合、システムに調整すべき一対比較値を明確化するために用いられる。ルール“comparison1”は次のように記述される。

```

comparison1:
if matrix(name=X,status=making,size=S,element=(E > 0))
then
modify(1,[element=compute(E-1)])&
column_row(S,E,C,R) &
qa_weight(X,C,R,W,Default) &
(Default == assumption,
add_assumption(matrix_element(name=X,column=C,
row=R,weight=W)))
add(matrix_element(name=X,column=C,
row=R,weight=W)))
  
```

ここで、ルールのRHSはPrologのゴールの並びとして解釈される。述語column_rowは行列のサイズSと要素番号Eから求める一対比較行列の要素の場所C行R列を求める。要素番号Eは一対比較行列において求めるべき要素の場所を表す番号である。述語qa_weightは、一対比較行列XのC行R列の一対比較値Wとそれに対するユーザの信念Default（確実な値(fact)もしくは仮の値(assumption)）を得て、その結果をデータベースに保存する（例えば、Prologの述語assertを利用する）。もし、データベースの中に一対比較行列XのC行R列の情報が既にあれば、述語qa_weightはその値を返す。これにより、後戻りした後に再度述語qa_weightが呼ばれた場合（既に決定された一対比較値を参照することにより）、ユーザが繰り返して一対比較をすることを回避する。RHSにおける4番目のRHSアクションはPrologのOR構造と同様に解釈される。この例では、述語qa_weightが返す変数Defaultの値が“assumption”ならば述語add_assumptionが呼ばれ、さもなければ述語addが呼ばれる。述語add_assumptionは述語qa_weightから得た一対比較行列XのC行R列の一対比較値Wを仮定としてWM（Working Memory）に追加し、データベースにおけるC行R列の一対比較値W及び信念Defaultを更新する。同様に、述語addは述語qa_weightから得た情報を事実としてWMに追加し、データベースにおけるC行R列の一対比較値及び信念を更新する。

ルール“comparison_check”は、必要な一対比較が全て終了したかどうかをチェックする。ルール“check”は非整合度尺度I.R.の値を決定するために用いられる。

3. 2. 2. TMS機能の利用

AHPに関連した類似の研究として、スプレッドシートを用いることによりAHPの作業を効率化しているものがある〔権藤89〕。ここでは、非整合度尺度I.R.の値が0.1を超えたとき、どの項目間の一対比較が原因となっているかを見いだすことはユーザの責任に任されている。この場合、感度分析に相当する手法を用いて非整合度尺度I.R.の値が小さくなるように一対比較の重みを修正する必要がある。また、一対比較の

順番は、ユーザに任されており、どの評価基準について一対比較を終了し、どの評価基準についてまだ一対比較が残されているかを迷うことがあり、そのための工夫が必要となる。

一方、本研究のアプローチでは非整合度尺度I.R.の値が0.1を超えた場合、その原因がどの項目間の一対比較にあるかを自動的に見だし修正するためにKORE/IEのTMS機能を用いる。TMSを用いた利点は、3.2.1節で述べたように、一対比較の答えにユーザの信念を反映できることにある（自信のない一対比較の答えを仮の値とすることができる）。また、一対比較の順番はルールにより動的に管理されており、システムがユーザを誘導するかたちで一対比較のための過程が実行される。一対比較の順番に関する本アプローチの利点は、3.1節で述べた一対比較における要素間の独立性やレベル分けのための作業を動的に管理できる点にある。さらに、本論文では言及していないが、Harkerの方法[16]を用いることにより、システムがいくつかの冗長な一対比較をパスしたり、一対比較毎にユーザから得られる答え（重み）の整合性を動的にチェックすることができる。

3.2.2.1. 矛盾解消手続き

図5においてルール“consistency”は、非整合度尺度I.R.の値が0.1よりも大きくなった場合、手続き“contradiction”を起動させる。手続き“contradiction”は矛盾を発見・解消するために用いられる手続きであり、KORE/IEの組み込みの手続きとしてTMS(Truth Maintenance System) [Doyle 79] に相当する機能を提供する。重み付けに関連したここでの矛盾は、非整合度尺度I.R.の値が0.1よりも大きくなったことである。矛盾の解消は、仮定（ここでは、ユーザ指定した“仮の一対比較値”）を修正していくことにより、非整合度尺度I.R.の値を0.1よりも小さくすることに相当する。ここで用いられる矛盾解消手続き及びその特長は文献[新谷 88a, Shin-tani 89]に詳しいが、概略を示すと図6のように示される。

ここでは新たにステップ1-1が追加されている。ステップ1-1は、矛盾の原因となる仮定をチェックする際の順番を指定するためのものである。ステップ1で得られる仮定のリストLはTMSにおけるnogood-setに相当する。TMSでの矛盾解消手続きでは、nogood-setにおける仮定は任意の順（実際は、TMSノードの依存関係に基づく深さ優先の探索）でチェックされる。一方、本手続きのステップ1-1による特長は、問題に依存して（ステップ1-1で示すように）仮定のチェックの順番を指定することである。これにより、矛盾を効果的に解消することが可能になる。ここでの効果的な矛盾解消とは効果的に非整合度尺度I.R.の値を小さくすることである。

- ステップ1： 矛盾を導き出した推論連鎖において、矛盾の原因となる仮定のリストLを求める。そして、ステップ1-1へ行く。

ステップ1-1： リストLを指定された手続きでソートする。そして、ステップ2へ行く。

ステップ2： もし、リストLが空リストなら、矛盾解消手続きは失敗となり、手続きを終了する。さもなければ、ステップ3へ行く。

ステップ3： リストLの先頭の仮定 L_{head} （つまり、 $L = [L_{head} | L_{tail}]$ ）がある推論ステップまで（推論連鎖をたどって）推論サイクルを戻し、仮定 L_{head} の否定を宣言する。そして、ステップ4へ行く。

ステップ4： もし、ステップ3の処理後の新たな推論連鎖の結果、矛盾が発生したなら、ステップ3の宣言を無効にして、 $L = L_{tail}$ としてステップ2へ行く。そうでなければ、ステップ5へ行く。

ステップ5： 矛盾解消手続きは成功し、手続きを終了する。

図6 矛盾解消手続きの概略

3.2.2.2. TMS機能の起動

ルール“consistency”は、次のように記述される。

```
consistency: if matrix(name=X, inconsistency_ratio > 0.1)
then
  (contradiction(Assumption, strategy),
  ;
  otherwise(Assumption, strategy, new_weight)).
```

ここで、RHSにおける述語“contradiction”は前節で述べた手続き“contradiction”に相当する。第1引数は手続きが失敗した際に（図6のステップ2の場合）、検査した仮定のリストを返すために用いられる。第2引数で示した手続きstrategyは、矛盾を生じさせた仮定（つまり、仮の重み付け）の候補が複数ある場合に、検査する順番を決めるためのものである。この手続きは、Prologの述語としてユーザにより（アリティが1として）定義され、図6のステップ1-1で用いられる手続きであり、リストLの要素をソートするためのものである。具体的には、手続きstrategyは、3.2.3節で述べる重みを修正するために決定される重みの増減の量の大きい順にリストLをソートする。第2引数が指定されない場合は、図6におけるステップ1で得られた仮定リストに並んだ順（つまり、推論連鎖の深さ順）に仮定が検査

される。

ルールconsistencyでは、もし手続きcontradictionが矛盾の解消に失敗したら（図6のステップ2の場合）、述語otherwiseと呼ばれる。述語otherwiseは、手続きcontradictionと同様にKORE/IEの組み込みの手続きであり、推論を取り消すことにより矛盾の解消を図るが、一度に複数の仮定を修正する。述語otherwiseにおける第3引数のnew_weightは、3.2節で示すルールchange_weightのRHSで用いられる一対比較値を具体的に変更するための手続き名（述語名）である。具体的には、手続きotherwiseは、ソート手続きstrategyを用いてソートされた仮定リストAssumptionのリストから複数の仮定を取り出し、これら取り出された仮定を同時に修正することにより矛盾の解決を図る。仮定の取り出し方は、先ず最初は先頭から2つを取り出し（1つの場合は既に手続きcontradictionで検査済みだから）、これら仮定の一対比較値を同時に修正する。もしその結果、矛盾がさらに起これば（非整合度尺度IR値が0.1より大きければ）、仮定リストの先頭から3つの仮定を取り出し、3つの仮定を同時に修正する。最終的には、仮定リストの全ての仮定を同時に修正する。

ここでの修正は、手続きcontradictionとは異なり、単に仮定の否定を宣言するのではなく、第3引数で指定した手続きを用いて直接に仮定の内容（ここでは、スロット記述の一対比較値）を更新する。これは、前向き型プロダクションシステムにおいて、同時に仮定を修正するには一回の認識-行動サイクル（すなわち、一回のルールの呼び出し）の間で行なわれる必要があるからである。つまり、手続きotherwiseはルールchange_weightの機能も包含していることに相当する。

3.2.3. AHP理論に基づく重み付けの修正

矛盾解消手続き“contradiction”では、図6のステップ3で示すように検査対象になった仮定の否定を宣言するのみであり、具体的に一対比較値を修正するための手続きが必要となる。ルールchange_weightは、この一対比較値を修正するためのものであり、次のように記述される。

```
change_weight:
  if \matrix_element(name=X,column=C,row=R,weight=W)
  then
    new_weight(X,W,W2) &
    make(matrix_element(name=X,column=C,
                        row=R,weight=W2)).
```

ここで、LHSにおけるパターンは否定パターンと呼ばれ、シンボル“\”を用いることにより、正のパターンの否定を表している。但し、否定パターンは負のパターンではなく、推論における扱いは正のパターンである。これは、単にパターンにおけるスロット記述の情報をシンボル“\”（前置子として定義されている）で表現しているのにすぎない。例えば、この例での負のパタ

ーンは次のよう記述したのと同じである。

matrix_element(proof=nil,name=X,column=C,row=R,weight=W)
つまり、シンボル“\”は、このパターンの第1引数のスロット記述である“proof=nil”を表現していることに相当する。

ルールchange_weightは、矛盾解消手続きがある一対比較値を否定することにより活性化され、その一対比較値を修正するために手続きnew_weightを起動する。手続きnew_weightは、ルールで示すように、一対比較行列XのC行R列の値Wを値W2に更新するために用いられる。値W2は、3.2節の最初に述べたように値Wを±2の範囲で微調整することにより得られる。増減の方向は、一対比較行列Xの非整合度尺度IRを小さくする方向に決定される。これは、式(1)の行列Xの最大固有値を小さくする問題に帰着する。しかしながら、一般にはこの種の問題を効果的に解くことが困難なことから、本システムではAHPで得られる一対比較行列の特徴から、修正のための増減の方向を決定する。

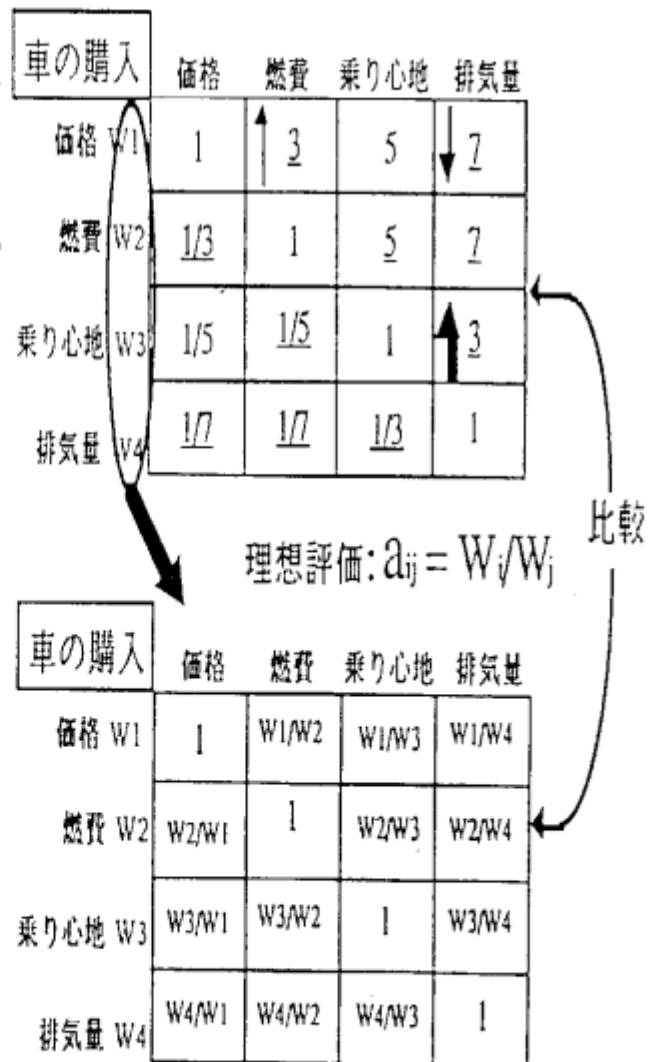


図7 一対比較値修正のための増減方向の決定

例えば、増減の方向の決定は図7を用いて説明できる。図7では2つ(上と下)の一对比較行列が示されている。一对比較行列(上)の行列要素で下線のある一对比較値は仮の重み付けを表しており、矢印の方向は修正のための増減の方向、そして矢印の太さは増減の量を表している。例えば1行4列の要素の一对比較値は仮の値として7が与えられ、一对比較の整合性を保持するために7より小さくする必要があることを示している。増減の量は、太さが2番目の矢印で示されているので中程度の減を行なう必要がある。増減の量は3つの矢印(大、中、小)を用いることにより表現される。

図7の一对比較行列(下)は、以上のような増減を決定するために用いられる情報であり、一对比較行列(上)の各項目に対する重み W_i を用いて求める。重み W_i は、3.1節で述べたように、一对比較行列(上)から得られる最大固有値に対する固有ベクトルとして得られる。一对比較行列(上)において i 行 j 列の一对比較値 a_{ij} は、項目 I_i の重みと項目 I_j の重みを一对比較して得られた相対的重み付けであり、理想的には W_i/W_j を求めたことに相当する。一对比較行列(下)の要素は、この W_i/W_j を用いる。このように構成される一对比較行列(下)の最大固有値は項目数 n と等しくなり、式(1)から非整合度尺度 $I.R.$ は0となる。つまり、一对比較行列(下)は理想的な一对比較を行なった場合として見なせる。そこで、この一对比較行列(下)に近付けるように一对比較行列(上)の各要素を修正することにより、非整合度尺度 $I.R.$ の値を効果的に小さくできる。先に述べた、一对比較値を修正するための具体的な増減の数値はこのようにして決定される。

重みの自動調整は、本節で述べた理想的な一对比較に近付ける方向に行なわれる。ここでの自動調整の妥当性は、基本的には、ユーザの責任において感度分析的な手法を用いてチェックする必要がある。しかしながら、実験的に次の3つの理由により、ユーザにとりかなり妥当な結果が自動的に得られている。第一の理由は、一对比較の一度に比較すべき要素数(最大7つ)を制限したことや要素間の独立性のチェックをすることにより、一对比較から得られる最初の結果がそれほど矛盾していないことにある。第二の理由は、TMS機構を用いることにより一对比較の答えにユーザの信念を反映でき、自信のない一对比較である仮の値を修正したことにある。第三の理由は、重みの自動修正を±2の範囲に制限したことにより、ユーザの許容範囲と思われる自動修正結果を得られたことにある。

また、一般に、TMS機構を用いる場合、解を得るのための収束性が問題になる。本アプローチでは、チェックすべき要素の数を最大7つに制限したことと、収束すべき上限として本節で挙げた理想的な一对比較行列を用いることにより、実用に耐える収束性を得ている。

4. おわりに

CDSSはESP(及びProlog)上のKORE/AEを用いてインプリメントされている。KORE/AEはOPS5風のルールプログラミング及び論理プログラミングの環境を提供する。CDSSの主機は、意思決定問題の複雑さを解消するために考慮すべき要素を階層的に整理し、要素の重要度を明らかにすることである。考慮すべき要素は、CDSSにおける階層設計支援機構やアウトラインプロセッサ機構を用いることにより視覚的に整理される。その際、KJ法などの利用が考えられ、本機構は情報を構造化するための機能を提供する。一对比較法は、意思決定者の主観的評価を簡便に取り入れることができる利点を有している。一方、その重み付けの整合性(すなわち、一对比較の推移律)を成立させるためには、要素間で煩雑な一对比較のやり直しを強いられるのが一般的である。CDSSでは、重み付けの整合性を保存するために、KORE/AEの非単調推論機能に基づくTMS機構を用いた。これにより、矛盾を含む一对比較は調整され、重み付けの整合性は効果的に保持される。

CDSSは、インタビューに基づく知識獲得支援ツール[溝口88]であるMORE[Kahn 85]、MOLE[Eshelman 86]やETS[Boose 84]の枠組み(ユーザに自分自身が持つ知識を意識させるための機能を追求すること)に類似している。これら知識獲得支援ツールの特徴は、問題領域として分類型のエキスパートシステムを想定し、問題領域に依存した知識の獲得のためのインタビュー戦略を用いることにより、診断型のルールを生成・洗練化することである。一方、CDSSの大きな特徴は、問題領域には依存しないで、対象とする問題を明らかにするために考慮すべき基本的な要素を整理し、これら要素に対する主観的評価を整合的に定量化することにより問題の解決を支援することである。今後の課題は、MORE/MOLE等実現されている問題領域のモデルを構築・洗練する機能を付加することにより、効率的な知識獲得支援システムを構築することである。

尚、本研究は第5世代コンピュータプロジェクトの一環として行なわれたものである。

謝辞

日頃よりご指導頂く当研究所戸田光彦研究員ならびに國藤進研究員に感謝いたします。本研究を進めるに当たり、貴重な御意見を頂いた当研究所片山佳則研究員及び、平石邦彦研究員に深く謝いたします。

参考文献

- [Boose 84] J.H.Boose: "Personal construct theory and the transfer of human expertise", Proc. of AAAI-84, pp.27-33, (1984)
- [Doyle 79] J.Doyle: "Truth Maintenance System", Artificial Intelligence Vol.12, pp.231-272, (1979)
- [Eshelman 86] L.Eshelman, et al.: "MOLE: A knowledge acquisition tool that uses its head", Proc. of AAAI-86,

pp.950-955,(1986)

[Forgy 81] C.L.Forgy: "OPS5 User's Manual", CMU-CS-81-135, July,(1981).

[権藤 89]権藤 他: ロータス1-2-3によるAHPシート, オペレーションズ・リサーチ, Vol.34 No.4, 1989.

[Harker 87] P.T.Harker: "Incomplete pairwise comparisons in the analytic hierarchy process", Math. Modelling, 9, pp.838-848,(1987)

[Kahn 85] G.Kahn, et al.: "MORE: An intelligent knowledge acquisition tool", Proc. of UCAI-85, 1, pp.581-584,(1985)

[溝口 88]溝口, 角所: 知識獲得支援システム, 人工知能学会誌, Vol.3, No.6, pp.732-740,(1988)

[Saaty 80] T.L.Saaty: The Analytic Hierarchy Process, McGraw Hill,(1980)

[中山 86]中山: 多目的意思決定-理論と応用-II, システムと制御, Vol.30, No.7, pp.430-438,(1986)

[新谷 86a]新谷, 片山, 平石: 問題解決支援環境KORE (その2) -知識記憶利用機構KORE/EDENとその応用-, 情報32回全国大会, 5L-9,(1986)

[新谷 87]新谷: 推論エンジンKORE/IE -反駁メカニズムに基づく高度な推論エンジン-, Proc. LPC'87, pp.233-242,(1987)

[Shintani 88] T.Shintani: "A Fast Prolog-Based Production System KORE/IE", Logic Programming: Proceedings of the Fifth International Conference and Symposium (edited by R.A.Kowalski and K.A.Bowen), MIT Press, pp.26-41,(1988)

[新谷 88a]新谷: アログクシオンシステムKORE/IEにおける非単調推論, Proc.LPC'88, pp.73-82,(1988)

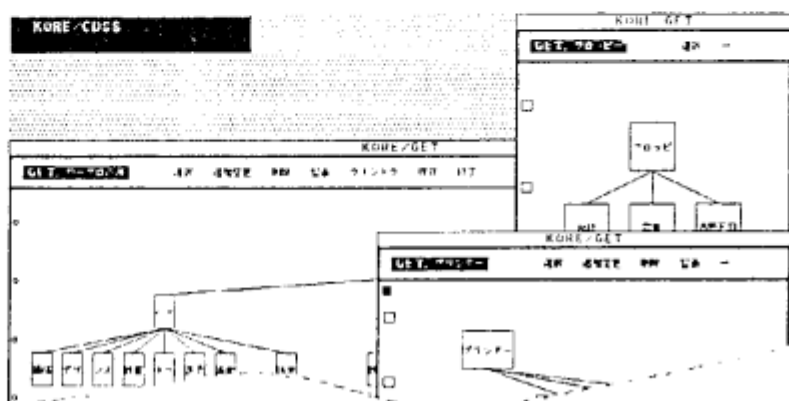
[新谷 88b]新谷: 問題解決支援機構KORE/CDSSにおける知識の構造化, 情報37回全国大会, 2J-5(1988)

[Shintani 89] T.Shintani: "An Approach to Nonmonotonic Inference Mechanism in Production System KORE/IE", LNCS Logic Programming '87, Springer-Verlag(1989).(to appear)

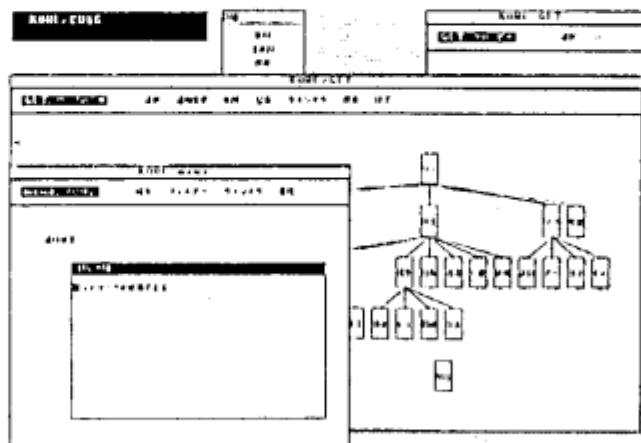
[付録] 実行例

本付録では, PSI上のCDSSの具体的な実行例を示す。

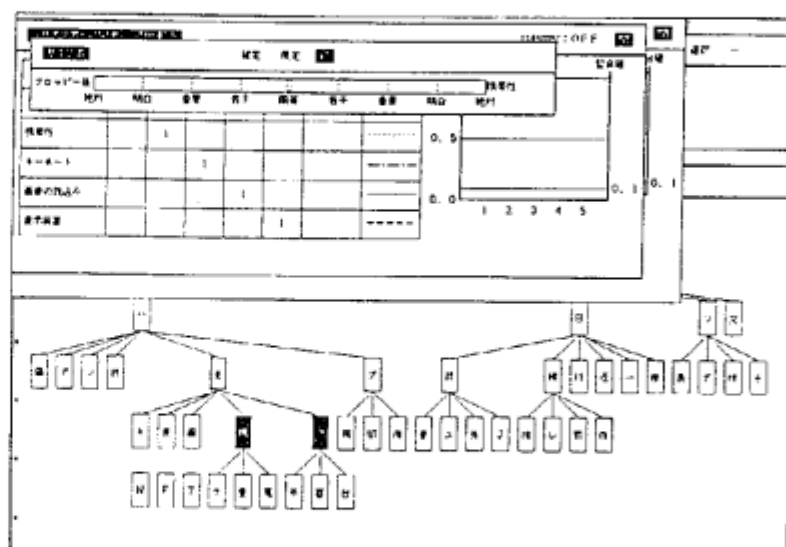
実行例(1): 階層設計支援機構の実行: 本例は, 第2章で述べた階層設計支援機構を用いて, 意思決定問題において考慮すべき要素間の関係を明らかにするために, 階層エディタにより問題の構造を階層的/視覚的に整理をしている様子を示している。この例では, "ワープロの選定"の問題(一番下のウィンドウで)に関連して, 既に整理済みの"プリンター"や"フロッピーディスク装置"の要求項目を張り込むために, それぞれのウィンドウを開いている。



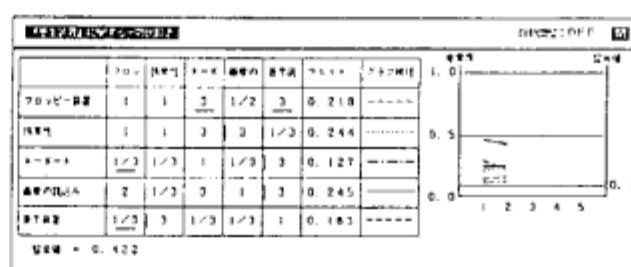
実行例(2): アウトライン・プロセッサ機構の実行: 本例は, 2章で述べたアウトライン・プロセッサ機構(一番上のウィンドウ)を用いて知識の断片を整理している例を示している。この例では, "通信機能"に関連して, "機能分類", "ネットワークが使用できる", "通信装置"という知識の断片を整理し構造化している様子を示している。構造化は, フォルダと呼ばれるウィンドウを用いて関係する要素をグループ化することにより実現する。ここでは, "機能分類"というキーワードをフォルダの名前とし, その中に"ネットワークが使用できる"という概念を入れることにより情報の構造化を図っている。フォルダ(つまり, グループ)化された概念は, CDSSにおけるカット&ペースト機能により, 階層設計支援機構の階層エディタに貼込まれるとグラフとして表現される。例えば, フォルダの名前が親ノードとなり, そこに含まれる要素は子ノードとなる。



実行例(3): 主観的評価支援機構における一対比較の実行: 本例は第3章で詳述した主観的評価支援機構における一対比較の実行例を示している。一番上のウィンドウは、「基本装置」に関連して「フロッピーディスク装置」と「携帯性」がどちらかがどれだけ重要かを一対比較により決定するための入力ウィンドウを示している。もし、「携帯性」の方が若干重要ならば、入力ウィンドウの「携帯性」側の「若干」の部分をクリックすればよい。入力ウィンドウの上部で示される「確定」及び「仮定」は、3.2.1節で述べられた一対比較値に対する信念（「確実な値」か「仮の値」かの区別）を表明するために用いられる。上から2番目のウィンドウは一対比較行列を表示している。



実行例(4): 主観的評価支援機構における重み付けの修正: 本例は、3.2.3節で述べた重み付けの修正過程を示している。重みの修正は、もし自調整ボタン（ウィンドウ上部右）がONになっていれば、3.1節で述べた非整合度尺度の値を0.1以下にするまで自動的に行なわれる。修正過程は本ウィンドウの右側のグラフで表示される。



実行例(5): 重要度の表示: 本例は、いくつかの一対比較行列から得られる重要度を合成することにより、最終的な各項目の重要度の表示例を示している。本例では、「ワープロの選定」（「ワ」で示されるルートノード）に対する重み付けをサブノードに割り振ることにより（1.00を割り振る）、考慮すべき要素の重要度を表している（ここでは、1以下の数字は0で表示されている）。例えば、第2レベル（上から2段目）の項目間では「日本語処理機能」（「日」で示されるルートノード）がもっとも重要度が高くなっている。ウィンドウ下部左に並ぶノード「N」、「F」、「T」は選択すべき会社を表している。この例からは、重み付けが最も大きい「N」社を選択することが望ましいことになる。

