TR-644

A Rule based Consistency Maintenance for Subjective Judgements

by
T. Shintani (Fujitsu)

May, 1991

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

(03)3456-3191 ~5
Telex ICOT J32964

Institute for New Generation Computer Technology

A Rule based Consistency Maintenance for Subjective Judgments

Toramatsu SHINTANI

International Institute for Advanced Study of Social Information Science
FUJITSU LIMITED
17-25, Shinkamata 1-Choume, Ota-ku, Tokyo 144, Japan
(Phone : +81-3-730-3111)
E-MAIL: tora@iias.fujitsu.co.jp

The Track : Artificial Intelligence

ABSTRACT

We propose a mechanism for maintaining consistency of subjective judgments by a decision maker. In the mechanism, subjective judgments are extracted from a decision maker, and they are used to select alternatives effectively. The subjective judgments are quantified by using AHP based pairwise comparisons. By using AHP method, a decision maker can perform almost valid comparisons. In order to maintain consistency of the comparisons, we exploit the TMS based nonmonotonic inference in the Prolog-based production system KORE/IE. By using the TMS based mechanism, a decision maker can represent his belief for the comparisons. The decision maker can group the comparisons into two classes such as (1)temporary(or assumption) comparisons and (2)reliable(or fact) comparisons. If an inconsistency has occurred, it can be effectively resolved by focusing on the temporary comparisons.

1. Introduction

Knowledge based systems can support decision makers. However, it is necessary to construct the knowledge base beforehand, and to be effective, it is also necessary to modify them through the decision process. In general, it takes a lot of time to assemble and codify knowledge in the knowledge bases, and such works are also tedious. As an alternative means of articulating knowledge beforehand, we implement a system which extracts and uses dynamically information of a decision maker in a decision process. The system is called CDSS(Choice Design Support System). CDSS is implemented on the production system KORE/IE(Shintani 1988). KORE/IE is a fast production system on Prolog. The inference mechanism is realized by performing a recognize-act cycle in a similar manner as OPS5(Forgy 1981) in which the type of reasoning is forward chaining.

CDSS supports a decision maker by structuring his knowledge and extracting his subjective judgments. Concretely, CDSS helps a decision maker make a reasonable choice from alternatives. In the choice process, two processes are used. One is a process which clarifies attributes used for estimating the alternatives. The other is a process which determines the priorities of the alternatives by estimating the attributes. In the estimation, a pairwise comparison method based on AHP (Analytic Hierarchy Process)(Saaty 1980) is used. AHP is one of the weighting methods, and has an advantage of being able to treat subjective judgments of a decision maker . However, in the method, it is difficult to maintain the consistency of the judgments.

In this paper, especially, we focus on the mechanism for maintaining the consistency and show the rule based method for constructing the mechanism in CDSS.

2. The outline of CDSS

Fig.1 illustrates processes for decision support in CDSS. CDSS supports a decision maker to create goals and their subgoals for a given problem. In the process, CDSS uses subjective judgments by a decision maker, and offers functions to clarify a structure of the problem. Knowledge of the decision maker is hierarchically structured through the process. In the structure, a problem(i.e. a goal) is hierarchically broken down into its smaller constituent parts(i.e. subgoals) which should be considered for solving the problem. The knowledge structured like this will be used as basic information for knowledge acquisition tools.

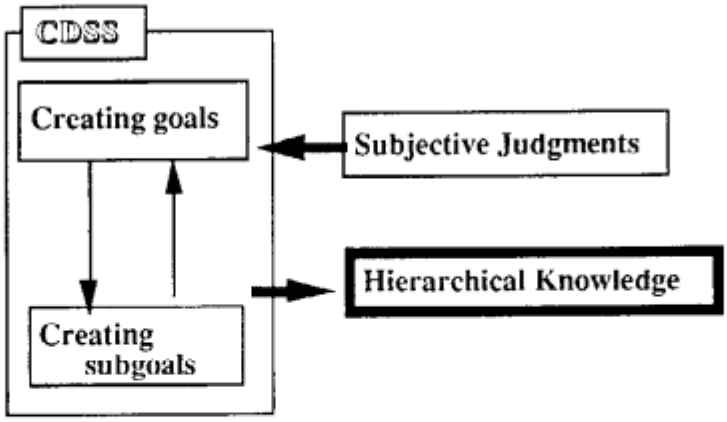

Fig.1. The Processes for decision support in CDSS

CDSS is composed of (1)the Hierarchy Design Support Subsystem(HDSS), (2)the Subjective Judgments Support Subsystem(SJSS), and (3)the Outline Processor Subsystem(OPS) as shown in Fig.2. The decision support processes in CDSS are achieved by combining and using these subsystems. HDSS supports to hierarchically decompose a given problem into its constituent parts which are elements for clarifying the problem. The complexity in the problem is canceled through extracting relations among the elements and arranging the elements hierarchically.

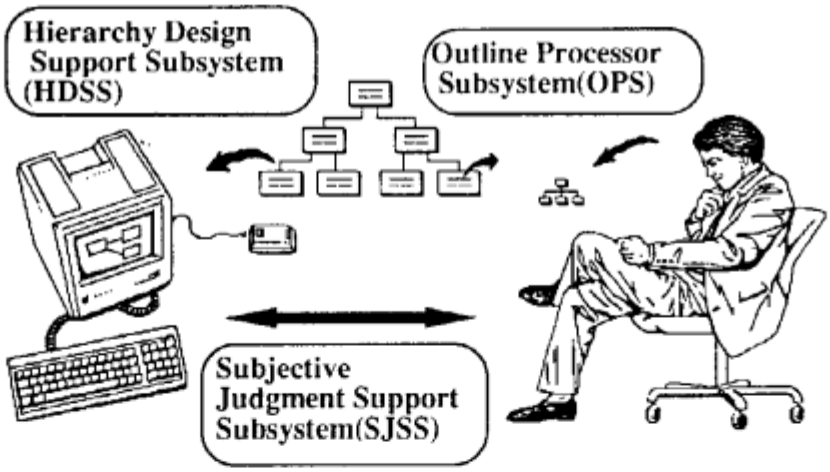

Fig.2. The subsystems in CDSS

The hierarchy is treated visually as a graph by using a graph editor which is named GET (Graph Editing Tool) as shown in Fig.3. The editor provides functions for manipulating the graph. The functions of the editor include (1)editing nodes and arcs of the graph, and (2)drawing the graph in a visually well-organized form which enables us to easily grasp the structure of the graph. In the example illustrated in Fig.3, the graph is used for clarifying superiority or inferiority among alternatives at the third level by enumerating and estimating the elements (at the second level) which should be considered concerning a goal. The goal is shown at the first level of the graph.

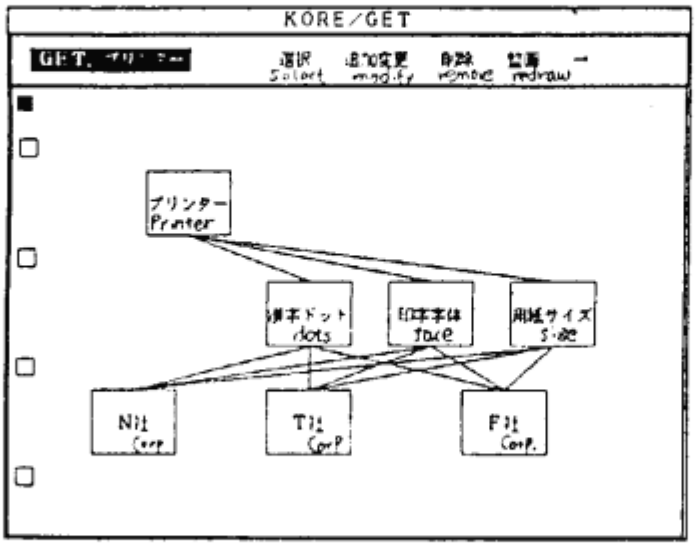

Fig.3. An example of the graph (in Japanese)

SJSS is used to quantify subjective judgments of a decision maker by using AHP based pairwise comparisons. SJSS provides a mechanism for maintaining the consistency of the pairwise comparisons automatically. The maintenance mechanism is realized by using rule based programming and a nonmonotonic inference function in KORE/IE(Shintani 1989). If inconsistency has occurred, inadequate pairwise comparisons are checked and revised first for maintaining the consistency. The maintenance mechanism is discussed in more detail in Section 3.

OPS supports a decision maker to arrange and structure his knowledge for crarifying a given problem. The organized knowledge is used and visually clarified as basic information for the decision support process in HDSS. CDSS provides a simulation function in the decision process by using HDSS and SJSS cooperatively. The function corresponds to a sensitivity analysis which is used to see how changing a judgment has an effect on a conclusion. Through the simulation, the decision maker gets information to evaluate and organize knowledge for the decision process.

3. The rule based consistency maintenance

3.1. Quantifying subjective judgments

A decision maker can choose the better between elements of a problem by using his ratio scales which are obtained from his subjective judgments by using AHP based pairwise comparisons. The comparisons are used to determine the relation importance of the elements in a level with respect to the elements in the level immediately above it. A

matrix is set up by carrying out the comparisons. The matrix is called the pairwise comparison matrix. The pairwise comparison matrix is used to generate the ratio scale for clarifying the priority of the element in the level. The pairwise comparison matrix is a square matrix whose row (or column) consists of elements used for the comparisons. The content of the matrix is weighting numbers for the comparisons where if the value V for element a_{ij} is determined then the reciprocal value 1/V is automatically entered in a_{ji}. The value V takes an integer which is less 9. The larger the number, the more importantly the judgment is agreed upon.

In AHP, the consistency can be checked by using the inconsistency ratio(I.R.) which is defined as follows;

$$(\lambda_{max} - n)/(n - 1)$$

where λ_{max}, n are the maximum eigenvalue and the size of the pairwise comparison matrix, respectively. If the I.R. is less than 0.1, the consistency is considered acceptable; otherwise it is inconsistenct. If the inconsistency has occurred, we need to revise a few pairwise comparisons for maintaining the consistency by finding the inadequate comparisons. However, it is difficult to find the comparisons. Usually, in the worst case, it forces us to revise all the comparisons.

3.2. The process for pairwise comparisons

Fig.4 illustrates the process for pairwise comparisons in SJSS. In Fig.4, the symbol encircled by a square represents a name of a rule. The rule "*start*" is used for initializing a state for pairwise comparisons.

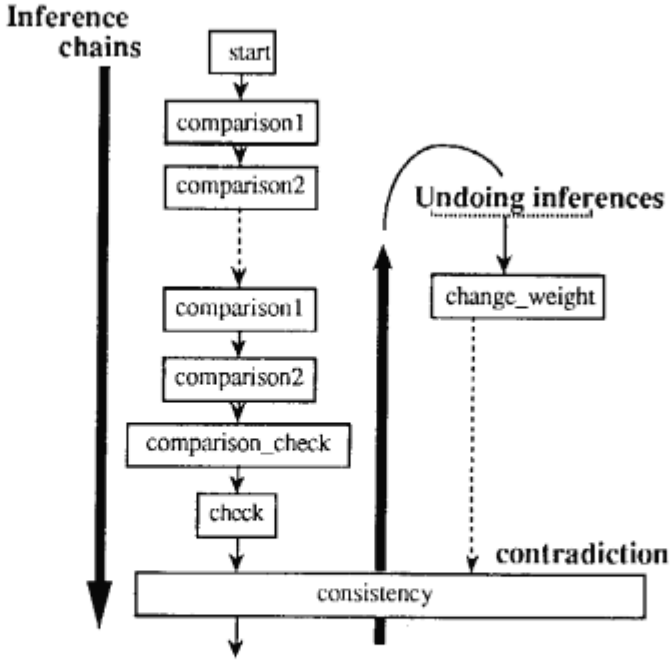

Fig.4. The Process for pairwise comparisons in SJSS

The rule "*comparison1*" and "*comparison2*" are used for obtaining a pairwise comparison matrix. The rule *comparison2* determines and enters the reciprocal values into the matrix. The rule *comparison2* means that "*If the value of the cell in the C-th column, the R-th row of the matrix X is the weight W, and the value of the cell in the R-th column, the C-th row of the matrix X is undefined, then put the weight 1/W into the cell in the R-th column, the C-th row of the matrix X*". The rule *comparison2* can be described as follows;

```
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))).
```

where a rule in KORE/IE consists of (1)the name of the rule (such as *comparison2*), (2)the symbol "**:**", (3)the symbol "**if**", (4)the LHS(Left Hand Side) which has some LHS patterns (such as "*matrix_element(name=X, ...)*"), (5)the symbol "**then**", (6)the RHS(Right Hand Side) which has some RHS actions (such as "*add*"), and (7) the symbol "**.**". The symbol "**&**" is a delimiter. The patterns correspond to compound terms of Prolog. The arguments are a sequence of one or more slot-value pairs. The pairs correspond to terms which have functions of arity 2 (e.g. =, \==, <, >). The second LHS pattern with the symbol "**-**" represents a negative pattern. In a matching process, the negative pattern is satisfied if no working memory(WM) elements match the pattern. The action *add* is used to create a new WM element and put the weight *W* into the cell in the *R*-th column, the *C*-th row of the matrix *X*.

By the rule *comparison1*, weighting numbers are inserted and grouped into the matrix. The rule *comparison1* can be described as follows;

```
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))).
```

where the execution of RHS actions correspond to that of Prolog goals. The fourth RHS action is executed based on the OR control structure in Prolog. In the execution of the action *qa_weight*, pairwise comparisons are obtained, and a decision maker declares his belief for the pairwise comparisons. The fifth argument *Default* of the action *qa_weight* returns the status of the belief. The action *qa_weight* keeps the information of the comparisons in a database. The information is used to avoid the comparisons which had been already performed previously when the rule will be triggered by the nonmonotonic inference mechanism in KORE/IE. If a cell in the *C*-th column, the *R*-th row of the matrix *X* is already filled by the weight number *W*, the action *qa_weight* returns the number *W* with the belief *Default*. There are two type of the belief which are (1)an assumption and (2)a fact. Concretely, the decision maker can group the weighting numbers into two classes such as (1)temporary(or assumption) numbers and (2)reliable(or fact) numbers which can be generated by using the actions *add_assumption* and *add*, respectively. The actions are defined as predicates of Prolog. The action *add_assumption* is used to create a new assumption in the WM, and keep the temporary number W and the information of the cell in the database. If inconsistency has occurred, the temporary numbers are checked and revised first for maintaining the consistency.

The rule "*comparison_check*" checks whether the matrix is filled or not. The rule "*check*" is used for getting the I.R. of the matrix.

3.3. The mechanism for maintaining the consistency

If the I.R. comes to be greater than 0.1, SJSS finds the causes for the inconsistency, and revises the causes by using the TMS(Doyle 1979) based inference mechanism in KORE/IE. The causes correspond to inadequate pairwise comparisons. The TMS based mechanism provides some advantages as follows: (1)In order to realize flexible pairwise comparisons, a decision maker can represent his belief for the pairwise comparisons. (2)The system can dynamically manage the order of the comparisons for realizing efficient comparisons. For example, SJSS can omit some redundant pairwise comparisons and check the consistency of each comparison dynamically by exploiting the Harker's method(Harker 1987).

The rule "*consistency*" is used for achieving the action "*contradiction*" if the I.R. is greater than 0.1. The action is used to find an inadequate assumption and resolve an inconsistency by revising the assumption. The action provides a TMS based function for maintaining consistency among pairwise comparisons. In SJSS, by using the action, the inconsistency is resolved by decreasing a temporary number of I.R. in which the number will come to be less than 0.1.

The outline of the algorithm for the action *contradiction* can be shown as in Fig.5.

Step 1: Get a list L of assumption on which an inconsistency depends. Go to Step 1-1.

Step 1-1: Sort the list L by using a specified strategy. Go to Step 2.

Step 2: If the list L is empty then stop the action and the action becomes a failure; otherwise go to Step 3.

Step 3: Get an assumption from the list L, which is the first element(that is, L_{head}) of the list L (that is, $L=[L_{head}|L_{tail}]$), and backtrack to the inference step which includes the assumption. Then, assert a negation of the assumption. Go to Step 4.

Step 4: If a new inconsistency is detected, then undo the result of Step 3, let $L=L_{tail}$, and go to Step 2; otherwise remove the data which depend on the assumption, and go to 5.

Step 5: Stop the action. The action succeeds.

Fig.5. Sketching the algorithm for the action *contradiction*.

The list L corresponds to the nogood-set of TMS. In TMS, assumptions in the nogood-set are checked in an arbitrary order. The feature of the algorithm shown in Fig.5 is to provide a function to check the assumption in a specified order. By specifying the order the algorithm can resolve the inconsistency effectively. The rule *consistency* can be described as follows:

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

where the action *contradiction* is used in the RHS. If the action *contradiction* fails, the first argument of the action *contradiction* returns a list of assumptions which corresponds to the nogood-set. The second argument of the action is used for indicating a strategy which sorts the list of the assumptions. The strategy is defined as a Prolog predicate which has an argument. The strategy is used at Step1-1 in Fig.5. In SJSS, by using the strategy *decrease_IR*, the assumptions in the list are arranged in decreasing order of the amount of number which is increased and decreased for revising the temporary numbers. If the second argument is not indicated, the list obtained at Step 1 is directly used in which the assumptions are arranged in order of generating them.

In the RHS, if the action *contradiction* fails, the action otherwise is activated. The action *otherwise* resolves inconsistency in the same manner as the action *contradiction*. However, there are two main differences between the actions. One is that the action *otherwise* is used for revising some assumptions at the same time. Second is that the

action *otherwise* directly revises the temporary numbers in the list *Assumption* by using a specified procedure which is indicated at the third arguments. The reason for the direct revision is that in a forward chaining production system the simultaneous revision of the numbers corresponds to achieving the revision during each recognize-act cycle. Namely, the action *otherwise* includes a function for the rule *change_weight* mentioned in the next section. The simultaneous revision needs to be achieved only when the rule *consistency* is firing.

3.4. Revising the pairwise comparisons

A rule is needed to concretely revise an assumption because the action *contradiction* only negates the assumption by asserting the negation of the assumption as shown at Step 3 in Fig.5.

The rule "*change_weight*" is used for concretely revising the assumption(that is, an inadequate pairwise comparison), which can be described as follows;

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

where the LHS pattern corresponds to the negation of the inadequate comparison which needs to be revised. The negation is generated by the action *contradiction* if the action finds the comparison as an assumption. By firing the rule *change_weight*, the inadequate comparison is revised. In order to revise the comparison, the RHS action "*new_weight*" is used. The action changes the old weighting value "*W*" to the new weighting value "*W2*" which contributes to decrease the I.R. of the matrix. The range of the change takes ±2 because we assume that the old value is not so inadequate for the consistency. If the I.R. comes to be less than 0.1, the action *contradiction* stops and the consistency is automatically maintained.

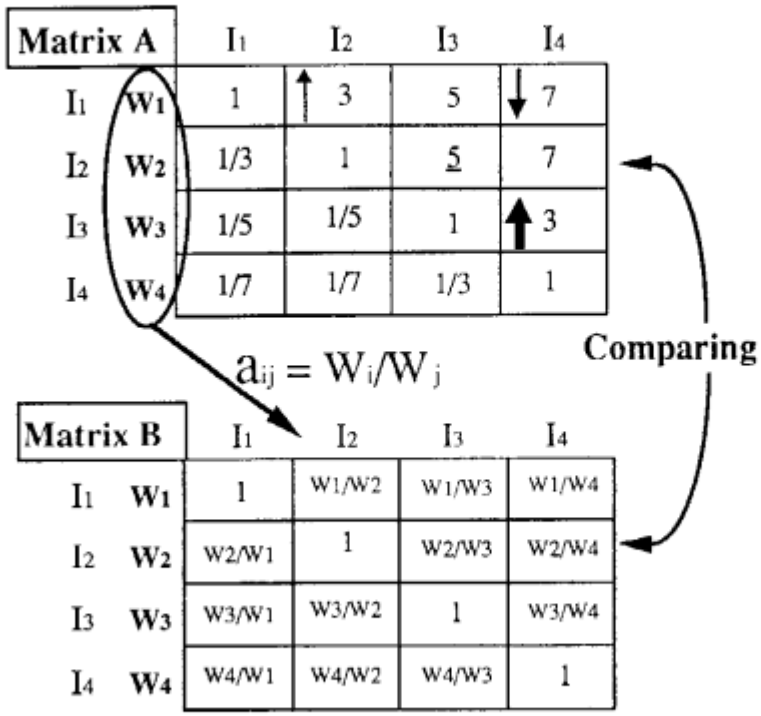

Fig.6. Setting up Matrix B from Matrix A

It is well known that it is difficult to determine the range of the change in which we need to decrease the maximum eigenvalue of a pairwise comparison matrix mentioned in Section 3.1. In order to concretely determine the range, we exploit the feature of the matrix based on AHP. Fig. 6 shows how the range is determined. It shows two pairwise comparison matrices which are Matrix A and Matrix B. An element a_{ij} in Matrix A is obtained by performing a pairwise comparison between elements I_i and I_j. In Matrix A, the direction of the arrows are used to represent the range of the direction in which the upper and down arrows indicate increasing and decreasing, respectively. The amount of increasing and decreasing is represented by using three kinds of arrows(that is, thin, normal, and thick) in which the thicker the arrow, the more the change is taken.

In order to determine the arrows, we set up Matrix B from Matrix A as shown Fig.6. Namely, Matrix B can be set up by using the weight of each element, which is computed from Matrix A. By exploiting the theory of AHP, the element a_{ij} in Matrix B are defined as W_i/W_j in which W_i and W_j are weights of the elements I_i and I_j in Matrix A, respectively. The I.R. of Matrix B comes to be 0 by using the method based on AHP. Matrix B corresponds to the ideal matrix in which pairwise comparisons of any problem are consistently performed. So, by gradually bringing Matrix A close to Matrix B, the I.R. of Matrix A will decrease effectively. Namely, the range of the change mentioned above can be determined by comparing Matrix A and B.

4. Conclusions

Subjective judgments of a decision maker are quantified easily by using the pairwise comparison method. In the method, it is difficult to maintain the consistency among the comparisons. In order to maintain the consistency, tedious revisions for the comparisons are required. In CDSS the consistency is automatically maintained by using the TMS based inference mechanism in KORE/IE. Generally speaking, the validity of the maintenance is to be checked by a decision maker. However, the decision maker can get a reasonable result after the automatic maintenance. The reasons of the results can be summarized as follows: (1) By using AHP method, a decision maker can perform almost valid comparisons. The decision maker can assume that the initial comparisons is not so inadequate for the consistency. (2) By using the TMS based mechanism, a decision maker can represent his belief for the comparisons. If the inconsistency arises, it can be resolved effectively by focusing on the assumption based comparisons. Generally, in case of using TMS, a convergence becomes a matter of concern. In our approach, the mechanism converges effectively to the reasonable result by using domain knowledge (that is, Matrix B mentioned in Section 3.4). (3) By restricting the range of the change (that is, ±2) for the comparisons, a decision maker gets reasonable result after the automatic revisions.

CDSS is similar to the frameworks of MORE(Kahn 1985), MOLE(Eshelman 1986) and ETS(Boose 1984) which are knowledge acquisition tools. These systems are used to obtain diagnosis knowledge from a decision maker and structure them by using a domain depend interview method. The main feature of CDSS is that CDSS supports a decision maker without the domain knowledge and the specified interview method. The future subject is that we enhance CDSS in order to realize a knowledge acquisition tool by adding functions for constructing and refining a model of a problem domain.

Acknowledgment

The author would like to acknowledge the continuing guidance and encouragement of Dr. Mitsuhiko Toda, IIAS-SIS. The author is deeply grateful to Mr. Susumu Kunifuji IIAS-SIS, for reading the draft of this paper and giving him many valuable comments.

This research has been carried out as a part of Fifth Generation Computer Project.

References

[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,(1985)

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

[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 IJCAI-85,1, pp.581-584(1985)

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

[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)

[Shintani 89] T.Shintani: "An Approach to Nonmonotonic Inference Mechanism in Production System KORE/IE",LNCS 384, Logic Programming'88, Springer-Verlag,pp.38-52(1989).