

CASE-BASED DECISION CLASS ANALYSIS FOR BUILDING A DECISION MODEL: ONION

Jae Kwang Lee

Department of e-Business, Korea Polytechnic University, jkleee@kpu.ac.kr

Jae Kyeong Kim

School of Business Administration, Kyung Hee University, jaek@khu.ac.kr

ABSTRACT

The decision modeling process requires much time and effort, and the resulting model, such as an influence diagram is applicable to only one specific problem. Existing studies reusing prior knowledge have some drawbacks, such as knowledge representation or lack of case problems. The basic idea of case-based reasoning is that humans reuse the problem solving experience to solve current problems. We suggest a case-based decision analysis (CB-DCA) methodology for building a decision model with influence diagrams. CB-DCA is composed of a case retrieval procedure and an adaptation procedure. Two measures are suggested for the retrieval procedure, one is a fitting ratio and the other is a garbage ratio. The adaptation procedure is based on decision-analytic knowledge and decision participants' domain specific knowledge. To implement CB-DCA, a prototype system named ONION is also developed. Evaluation results showed that ONION provides decision makers with an effective and efficient way for modeling decision problems with influence diagrams.

KEYWORDS: Decision analysis, Decision class, Influence diagram, Case-based reasoning

INTRODUCTION

The traditional formulation of real decision problems is done by lengthy interviews between decision maker (DM), domain expert(s), and decision analyst(s). Such a process requires much time, effort, and cost, but the main difficulty is that a constructed decision model such as influence diagrams (IDs) are usually applicable to only one specific problem(Olmsted, 1984). Decision makers and domain experts found that some prior knowledge from the experience modeling IDs can be utilized to resolve other similar decision problems(Kim, 1991). To reduce the burden of modeling decision problems, the concept of decision class analysis (DCA) was proposed by Holtzman(1989). DCA regards a decision analysis as an integrator of domain-specific knowledge and decision-analytic knowledge, and treats a set of decisions having some degree of similarity as a single unit. To implement DCA, domain-specific knowledge of a predefined decision class is stored at a explicit-typed or an implicit-typed knowledge base and it is retrieved to model a current decision problem. As a methodology to implement DCA, rule-based approaches(Chung, 1992; Kim, 1991; Reed, 1989), frame-based approaches(Sonnenberg, 1994), and neural network based approaches(Kim, 1997; Kim, 1998) have been used until now. Previous approaches are reported to give a relatively good performance in generating topological level IDs at restricted situations. However they have exposed some difficult issues regarding supporting decision modeling process, the number of cases, and knowledge acquisition. They do not provide for the decision participants what the decision participants do each step in decision modeling process. The difficulty of neural network-based approach is that it is not easy to get enough

similar decision problems for training the neural networks. Each decision problem and corresponding decision model is regarded as one element of the training set. Our experience shows that the number of similar decision problems in the strategic decision making area is not enough for training the neural networks. And rule-based approaches or frame-based approaches have difficult knowledge acquisition process.

To solve above difficult issues, we suggest a case-based decision class analysis (CB-DCA) methodology for modeling decision problems with IDs. The basic idea of case-based reasoning (CBR) is that humans reuse the previous problem solving experience to solve a current problem(Kolodner, 1991). CB-DCA regards an ID of one decision problem as a case, so it stores IDs of similar decision problems at a case base. Furthermore, CBR can acquire knowledge with ease using inductive methodology, so it is useful especially when knowledge is incomplete, or the number of cases are sparse(Kolodner, 1993). In addition, CBR works well in domains that are poorly understood, because the system doesn't need to know why something worked in the past(Kolodner, 1991). From these reasons, CBR is suggested to be used as a tool for modeling decision problems in the strategic decision making area.

The main task of using CBR is generally composed of the representation of a class, a retrieval procedure, and an adaptation procedure(Kolodner, 1991; Kolodner, 1993; Risebeck, 1989). In this research, case is represented as a frame-typed data structure corresponding to a decision situation and an ID. A retrieval procedure is suggested to retrieve one or more cases to model a current decision problem. To get an ID through modifying retrieved cases, an adaptation procedure is also suggested. To implement CB-DCA, a prototype system, named ONION is developed.

BACKGROUND

Influence Diagram

Influence diagrams (IDs) are developed as a model for representing complex decision problems based on incomplete and uncertain information from a variety of sources(Howard, 1984; Howard, 1988). The ID can be viewed from three levels: topological, functional, and numerical(Howard, 1988; Kim, 1999; Lee, 2000). At the topological or relational level, the nodes in the diagram represent the key variables in the system being modeled and the arcs or arrows identify conditional influences among them. The nature of these influences is specified at the functional level and further quantified at the numerical level. Each level provides a stage of decision making in a given domain.

At the topological level, the ID is defined as an acyclic digraph, with three types of nodes and two types of arcs. This visual level of the ID explicitly reveals the flow of information, influences, and overall structure of the decision problem. The rectangle symbolizes a *decision node* which represents a variable for the decision maker and contains alternatives to choose. The oval symbolizes a *chance node* which represents events and contains a random variable for the event. It contains probabilities assigned to the possible outcomes of the random variable. The rounded rectangle symbolizes a *value node* which represents an objective to maximize or minimize. An arc into a chance node implies a probabilistic dependency between the nodes. An arc into a decision node implies that when we make a decision, we have information on the value of the predecessor. Figure 1 shows a simplified example of an ID for a landfill expansion problem (Kim, 1997; Kim, 1999). In Figure 1, "Proposal permit or denial" is a decision node. Decision nodes represent decision options that are available to DM. It has branches to represent the possible options. Each chance node has an underlying probability distribution to quantify the uncertainty for the variable that node represents. Arcs into chance nodes represent information affecting the probability distribution for that node.

Chance nodes are generally represented as a cycle or an oval. “Utility” is a value node, and it summarizes the preferences of the DM for the outcomes. A mathematical function (value function or utility function) associated with value node can be used for deriving a numeric value representing the trade-off among attributes of the problem.

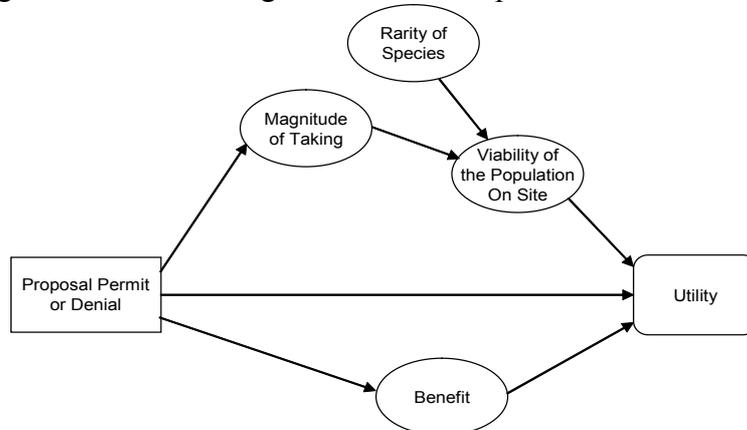


Figure 1. An ID for landfill expansion problem

The functional level is concerned with how nodes are related. At the final level, the numerical level, probability distributions, decision alternatives, costs, and the utilities of the DM are assessed numerically for each corresponding node. A well-formed ID (WFID) is a syntactically correct, completely assessed ID whose nodes have fully consistent distributions and outcomes (Holtzman, 1989). But in this research, we use well-formed ID to refer a well-constructed topological-level decision model from which a decision is made without further modification of the model.

The traditional interactive procedure to build an ID consists of a sequence of *value-preserving transformations* between domain expert(s), decision analyst(s), and DM (Holtzman, 1989; Kim, 1998). The value preserving transformation is a transformation of the ID which maintains feasibility and does not modify the optimal policy or maximal expected value. The interactive process to build an ID starts from a core ID and is expanded through the repetitive operation of adding nodes, and splitting nodes. Once the structure is reasonable, the diagram is further refined through the operation of node removal, merging nodes, and reversing arcs as well as adding and splitting nodes. The interactive procedure is known to be lengthy, and the resulting model is applicable to only one specific problem. It has been shown that these operations satisfy the value-preserving transformation (Chung, 1992; Kim, 1997; Shachter, 1986). Please refer appendix 2 about formal definition of ID and WFID.

Decision class analysis

The decision analysts have observed that a constructed decision model such as an influence diagram (ID) is usually applicable to only one specific problem, even if the formulation of a real decision problem needs much time, effort, and cost. They often investigate whether some prior knowledge from the experience of modeling IDs can be utilized to resolve other similar decision problems (Kim, 1992; Holtzman, 1989; Howard, 1988; Kim, 1998). Holtzman (1989) describes DCA as an integrator of decision knowledge and treats a set of decisions having some degree of similarity as a single unit. In the DCA, similar decision problems mean the problems in the same (sub)domain. DCA helps decision analysts to inexpensively model a decision problem from a cumulative set of decisions. Variables in the ID are changeable according to the *specific situations*. The specific situations may be decision nodes and decision maker's circumstances. These situations are called *situation frames* in subsequent

descriptions. A problem class consists of a number of individual decision problems, thus the size of a problem class is usually larger than that of each individual problem. When the situation-specific information is given, the DCA should abstract the corresponding specific decision variables for solving the individual problem. In this case, the DCA can be described as a *classification problem*. In this research, DCA is considered as a classification problem with a set of input data (situation-specific knowledge) and output data (a topological level ID). So the quality of resulting decision model depends on the quality of input data given from DM with the help of domain expert(s). Analyzing a class of decisions occurs at a higher level of abstraction than analyzing a single decision. Given the values of situation frames from the DM, the DCA should abstract and refine the corresponding specific decision variables for solving the individual problem.

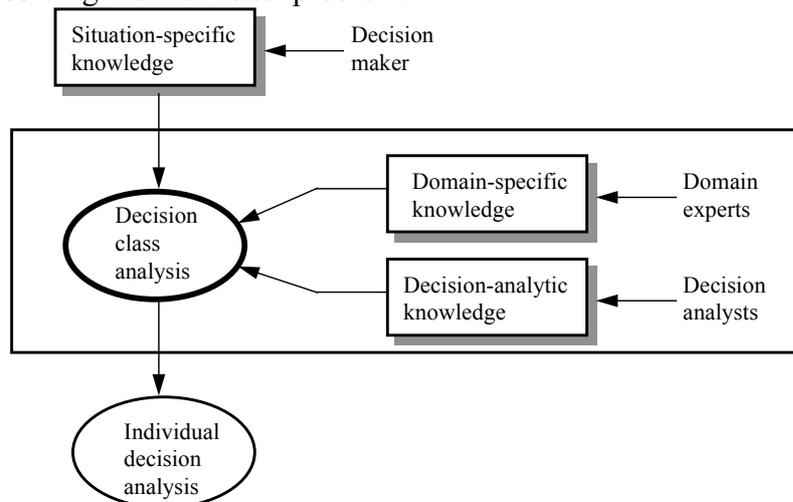


Figure 2. The knowledge about decision class analysis.

To analyze a class of decisions, rule-based approaches, frame-based approaches, and neural network based approach have been used as shown in Table 1.

Table 1. The researches for building decision models

Approaches	Researches	Problem domain	Decision model
<i>Rule-based</i>	Heckerman & Horvitz (1991)	Medical decision	Influence diagram
	Chung et al. (1992)	Buyer's decision	Influence diagram
	Egar & Musen (1994)	Medical decision	Influence diagram
	Kervahut & Potvin (1996)	Vehicle dispatching	Decision tree
	Sarkar et al. (1996)	Financial distress for banks	Belief network
<i>Frame-based</i>	Cooper et al. (1991)	Medical decision	Belief network
	Wen (1991)	Medical decision	Belief network
	Sonnenberg et al. (1994)	Medical decision	Influence diagram
<i>Neural-network based</i>	Kim & Park (1997)	Raw material purchasing	Influence diagram
	Machado & Campos (1997)	N/A	Influence diagram
	Kim et al. (1999)	Landfill expansion problem	Influence diagram

Rule-based approaches: Holtzman(1989) described the concept of decision class analysis using knowledge such as rules, and based on his concept, Kim(1992) developed a knowledge-based decision system (KIDS) to build an influence diagram from a class of decisions. Rule-based approaches tend to be domain-specific and function extremely well when decision problems are well defined. They are implemented by representing the domain experts' knowledge as a series of IF-THEN typed rules, which depend on the observation of the whole known combinations of experts' data. If the number of rules are not enough, and/or

associative, then rule-based approaches may be inappropriate (Kim, 1997; Kim, 1998). *Frame-based approaches*: Frame-based approaches use frames to represent complex type of knowledge. Sonnenberg et al (1994). suggested an architecture for knowledge-based building of influence diagrams in medical domain. They used frames to represent medical domain knowledge such as clinical events (diseases, complications, diagnostic tests, and treatments) and probabilistic dependency. *Neural-network based approaches*: Decision class analysis is viewed as a classification problem where a set of input-output data pairs is given. Decision situation representing decision problem is regarded as an input of a classification problem, and its corresponding topological level ID is regarded as an output. Kim and Park (1997) thus utilized a feed-forward neural network with a supervised learning procedure so as to generate an influence diagram in the topological level.

CBR

CBR means reasoning based on previous cases or experiences. CBR tasks are often divided into two types, *interpretation* and *problem-solving* (Kolodner, 1993). Interpretation CBR uses prior cases as reference points for classifying or characterizing current situation and problem-solving CBR uses prior cases to suggest solutions that might apply to current circumstance. The researches about case-based interpretation and case-based problem-solving are presented at Table 2.

Table 2. CBR researches

Types	Researches	Applications
<i>Interpretation</i>	Ashley (1990)	Legal system
	Porter et al. (1990)	Learning and classification
	Branting (1991)	Legal analysis
	Dayal et al. (1993)	Legal analysis
	Reategui et al. (1997)	Diagnosis in medicine
	Anandanpillai et al. (1999)	Housing discrimination law
	Pal et al. (2000)	Business acquisition
	Jarmulaka et al. (2001)	Data interpretation
	Chiu (2002)	Customer classification
<i>Problem-Solving</i>	Alexander et al (1989)	Design
	Hammond (1989)	Planning
	Tessem et al. (1997)	Software modeling
	Grupe et al. (1998)	Software design
	Suh et al. (1998)	Quality design
	Fdez-Riverola et al. (2003)	Forecasting
Luu et al. (2003)	Construction procurement	

Case-based Interpretation: In interpretive CBR, the reasoner's goal is to form a judgment about or classification of a current situation, by comparing and contrasting it with cases that have already been classified. For example, interpretive CBR plays a fundamental role in interpreting legal concepts and applying law in the legal system (Ashley, 1990; Chiu, 2002; Dayal, 1993). *Case-based Problem-Solving*: The goal of problem-solving CBR is to apply a prior solution to generate the solution to a current problem. For example, case-based design (Alexander, 1989) and planning (Hammond, 1989) systems retrieve and adapt solutions of similar prior problems.

The intuition of problem-solving CBR is that situations recur with regularity. What was done in one situation is likely to be applicable in a similar situation. This provides us with a basic motivation for using CBR as a main tool for storing previous decision models and reusing later for various decision-making problems. Case base of CBR can be used for a

reservoir for DCA. If problem specific knowledge is formalized in a form of multiple IDs, then they are stored in the case base and can be utilized later for doing decision-makings by retrieving and reusing them according to the characteristics of problems under consideration. CBR is a simple technique with a lot of intuitive appeal but also with a cognitive basis.

In this research, we suggest a CBR-based methodology to implement a DCA, i.e., to build a topological level ID. A case based reasoning approach is presented to effectively apply the past experiences and expertise of decision analysts for modeling an ID. A CBR has the following advantages for analyzing a class of decisions. CBR allows decision makers to propose solutions to problems quickly without need to derive those solutions from scratch. This provides organizational memory-based intuition for a given problem, which can avoid any irregular or abnormal problem-solving process. CBR can provide a systematic mechanism for storing domain-dependent knowledge and decision analytic knowledge as cases, and reusing them according to the characteristics of problems. Based on the past mistakes made by some decision makers in organization, CBR can alert decision makers to avoid repeating past mistakes. CBR can help decision makers point out what features of a problem are the important ones to remember during problem-solving.

CASE-BASED DCA METHODOLOGY

Decision making process using CBR-based methodology consists of three procedures including retrieval procedure, adaptation procedure, and problem solving procedure. The decision problem is formalized by an ID which is represented as a frame typed knowledge. A frame-typed knowledge is stored in the case base and prepared to be retrieved according to the characteristics of problems. Reuse of IDs is triggered by the presence of problems which decision makers want to solve. To facilitate the reuse process, we developed two procedures - *a retrieval procedure* and *an adaptation procedure*. The retrieval procedure contributes to choosing the appropriate IDs from the case base, and the adaptation procedure allows retrieved IDs to be properly updated to the changes in the environment so that the quality of the case base can be kept up-to-date. These two procedures generate the most suitable ID which best fits the current problem solving situation. The last problem solving procedure solves a given problem based on the result from the adaptation procedure.

Overall procedure

The overall procedure of suggested CBR-based methodology is depicted in Figure 3. Each step is explained as follows.

Step(1) Identifying the value of situation frame; DM input the value of situation frame of a given decision problem.

Step(2) Retrieving the candidate cases; The retrieval procedure selects appropriate cases from the cases which have same decisions with a given problem are selected. The selected cases are called candidate cases. The cases that consist of situations and corresponding IDs are compared with the situation of the current problem and are evaluated based on a similarity metric between the existing situation and current situation. If the existing case that exactly matches the current situation is found, its corresponding ID can be used without any modification. However, in many cases, several similar cases are chosen to be adapted for the current situation. The retrieval procedure is explained in detail in section 3.3. The adaptation procedure is carried out by two steps, combining candidate IDs and modifying the combined IDs.

Step(3) Combining candidate IDs. The retrieved candidate IDs are combined into two IDs,

super ID and core ID. The super ID is defined as the union of all candidate IDs, and the core ID is defined as the intersection of all candidate IDs.

Step(4) Modifying the combined ID. Supplementary nodes and arcs are defined as the difference between super ID and core ID. The modification process decides whether each supplementary node and arc is removed or not. The core ID with survived supplementary nodes and arcs results an ID which can be used as a decision model for a current problem.

Step(5) Building well-formed ID. Decision analytic knowledge is used to check and to correct whether the resulting ID is well-formed ID or not (please see appendix 2). If the ID does not satisfy the condition of WFID then DM modifies the ID to become a WFID, where optimal current decision is made without further modification of the model.

Step(6) Updating case base. Finally, the resulting WFID and the relevant situation are stored to the case base as a new case.

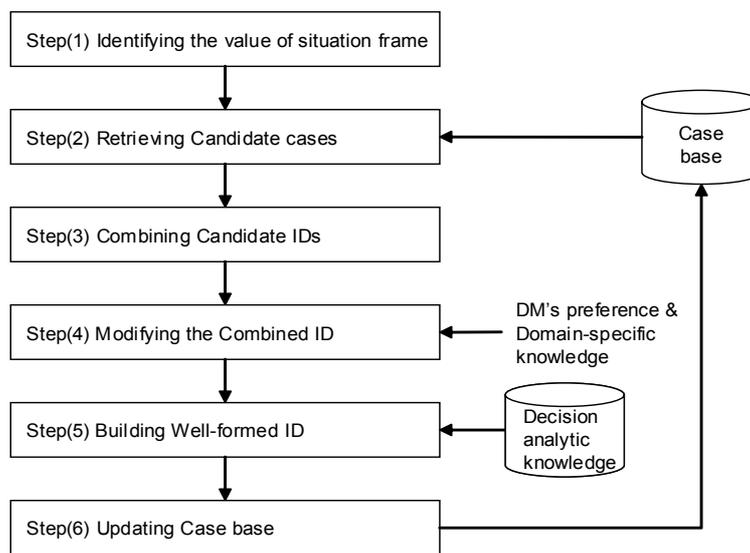


Figure 3. Overall procedure of suggested methodology

Case representation

To represent a case, we use a frame typed knowledge representation. A case is defined to be composed of an ID and its corresponding specific situation of one decision problem. In the terminology of DCA, a case is related with decision analysis, whereas a case base is related with decision class analysis. A case contains all the ID related information such as the situations of one specific decision problem, nodes, and arcs in the topological level. We focused on building a topological level ID, so it does not contain any functional and numerical leveled information(Kim, 1998).

```

{{Landfill-expansion-problem
IS-A : DECISION-PROBLEM
NAME : Landfill expansion problem
DATE : 20020530
DECISIONS : Proposal-permit-or-denial
SITUATION : (LE 1) (FC 0) (RE 0) (FA 0) (RC 0) (GC 0)
              (LQ 0.8) (WQ 0.8) (RA 0.9) (SP 0.9) (MO 0.3)
              (FF 0.2) (WE 0.8)
}}
  
```

(a) Example of decision problem

```

{{ Viability-of-the-population-on-site
IS-A : CHANCE-NODE
PART-OF : Landfill-expansion-problem
NAME : Viability-of-the-population-on-site
PREDECESSORS : Magnitude-of-taking Rarity-of-species
SUCCESSORS : Utility
}}
  
```

(b) Example of chance node

Figure 4. A part of case representation for landfill expansion problem

Topological level IDs of similar decision problems of a same class are stored in the same case base. The situation-specific knowledge about the decision problem plays a very significant

role in our approach. When retrieving a case from the case base, an important criteria is the degree of similarity between a given decision problem and the case of the case base. The frame-typed representation of an example case is shown like Figure 4 (a). The ‘DECISION-PROBLEM’ frame refers a specific decision problem and it contains the information about decision nodes and situation of the problem. An ID for a specific decision problem consists of decision nodes, chance nodes and value node. Each node contains node name, corresponding decision problem name, node type, predecessor node(s), and successor node(s). Figure 4 (b) shows a part of landfill expansion problem of Figure 1. The frame named ‘Landfill-expansion-problem’ has one decision node and thirteen situation frames. Each situation frame has a name and its value. The situation frame values are ranged from 0 to 1. Situation frames of the environmental review problems for land developments are shown at Table 3. A case base of one decision class consists of several ‘DECISION-PROBLEM’ frames like landfill-expansion-problem and their corresponding node frames like ‘Viability-of-the-population-on-site’. Whereas the ‘DECISION-PROBLEM’ frames are related with decision problems, the node frames are related with nodes of the IDs. The arc of an ID is represented by ‘PREDECESSORS’ and ‘SUCCESSORS’ of node frame. In Figure 1, ‘Magnitude-of-taking’ and ‘Rarity-of-species’ proceeds ‘Viability-of-the-population-on-site’ whereas ‘Viability-of-the-population-on-site’ proceeds ‘Utility’.

Table 3. Situation frames of environmental review problems for land developments

	Situation frame	Value
Problem situations	Landfill expansion problem (LE)	yes/no (0 or 1)
	Flood control problem (FC)	yes/no (0 or 1)
	Residence problem (RE)	yes/no (0 or 1)
	Factory problem (FA)	yes/no (0 or 1)
	Recreation problem (RC)	yes/no (0 or 1)
	Golf course problem (GC)	yes/no (0 or 1)
Environmental situations	Land quality of the site (LQ)	bad-good (0~1)
	Water quality of the site (WQ)	bad-good (0~1)
	Rarity of the site (RA)	low-high (0~1)
	Social pressures (SP)	low-high (0~1)
	Momentum of the project (MO)	small-large (0~1)
	Forest fragmentation (FF)	narrow-wide (0~1)
	Wetland (WE)	narrow-wide (0~1)

Case retrieval procedure

To reuse the ID stored in case base, we have to retrieve appropriate cases from the case base. One retrieved ID is usually not enough to represent current problem, so several relevant IDs, i.e., candidate IDs are retrieved. Next, the candidate IDs are unified and modified to be a WFID. In line of this argument, the basic idea of our suggested retrieval procedure is that the IDs representing the situations of a current problem are to be retrieved and at the same time the smaller number of retrieved IDs is better. For this purpose, we developed two measures, *the fitting ratio and the garbage ratio*. The information contained in situation is represented using situation frames, a set of situation-specific factors that have values ranged from 0 to 1. When a current problem is given, the fitting ratio measures the degree of coverage of the cases stored in case base. The more situation frames of the current problem are *covered* (or explained) by the case, the higher is the value of fitting ratio. Therefore, the fitting ratio indicates how much a case covers the situation of the current problem. On the other hand, the garbage ratio measures the degree of incongruity between the current problem and the case stored in case base. The more situation frames between the current situation and the case are

incongruent, the more they are incongruent to each other. Therefore, the garbage ratio measures how many situation frames of cases are incongruent with those of the current problem. The fitting ratio is used in a cumulative way - if one retrieved case is not enough for the explanation of the characteristics of current problem, several cases are selected in sequence according to the fitting ratio value. If all the situation frames of the current problem are explained by those of the retrieved cases, the current problem is said to be covered by the retrieved cases. The formal notation and procedure for the case retrieval are given appendix 1. The retrieval procedure selects a candidate case which has the maximum value of the ratio of fitting ratio to garbage ratio from case base. The retrieval procedure is iteratively performed until the current situation is covered over a predefined threshold value, θ_c ($0 < \theta_c \leq 1$). The retrieval procedure stops if the coverage of current situation is less than a predefined threshold value θ_c . Our heuristic experiences suggest a sufficient larger value to θ_c . If θ_c had a small value, retrieved candidate cases are lack for covering the current situation.

Case adaptation procedure

A case adaptation procedure builds an ID using the retrieved candidate cases. In Figure 2, two kinds of knowledge are necessary for the decision class analysis. Likewise, the adaptation procedure relies on decision-analytic knowledge and domain-specific knowledge of the decision participants. Decision-analytic knowledge consists of a set of model definitions. Model definitions are used to check whether the resulting ID is well-formed ID or not. The definitions for testing a constructed ID meets conditions of the well-formed ID are given appendix 2.

The adaptation procedure is presented as follows:

- Step(0) Define super ID and core ID form given candidate IDs. Define supplements nodes and arcs.*
- Step(1) Select a supplementary node from the super ID. If there are many supplementary nodes, choose one node arbitrary.*
- Step(2) Decide whether the selected supplementary node is accepted or not considering the level of analysis, the availability of information, and the characteristics of given problems. If supplementary node is not accepted, remove the supplementary node and its related arcs.*
- Step(3) If exists another supplementary node, then goto Step(1). Otherwise goto Step(4).*
- Step(4) Make the resulting ID to be well-formed ID, if it is not.*

The following example shows the modification process. Two candidate IDs as shown in Figure 5 are assumed to be selected by the case retrieval procedure.

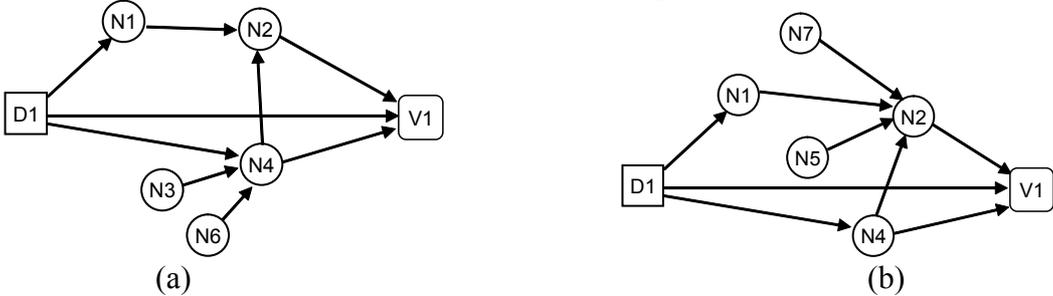


Figure 5. The candidate IDs

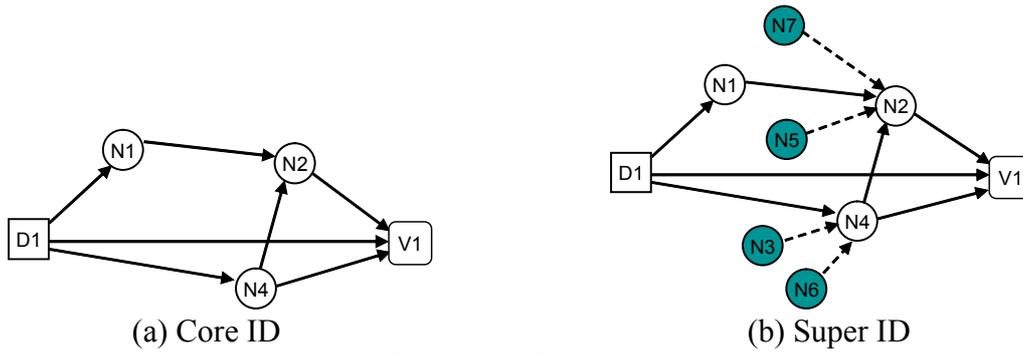


Figure 6. The core ID and the super ID

The core ID and the super ID is obtained as shown in Figure 6. In the super ID, N3, N5, N6, and N7 are supplementary nodes. Adaptation procedure is carried out until all the supplementary nodes are tested. DM has to decide whether all each supplementary node to be accepted or not respectively. Such a decision is dependent on the DM's preference, availability of information, characteristics of decision problems such as time pressure and cost. The ID in Figure 7 shows one possible resulting ID of the example.

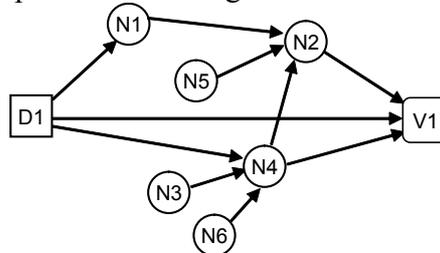


Figure 7. Resulting ID

PROTOTYPE SYSTEM: ONION

System architecture

To aid decision participants, the CBR-based methodology proposed in this paper is implemented as an ONION prototype system, named ONION. ONION consists of Case base, Knowledge base, and three major modules. The modules are composed of User Interface, Case base Manager, and ID Modeler. The modules of ONION are implemented using Visual C++. MS SQL 2000 for Windows is used to implement Case base and Knowledge base of ONION. Figure 8 shows the system architecture and the relationships among these system components.

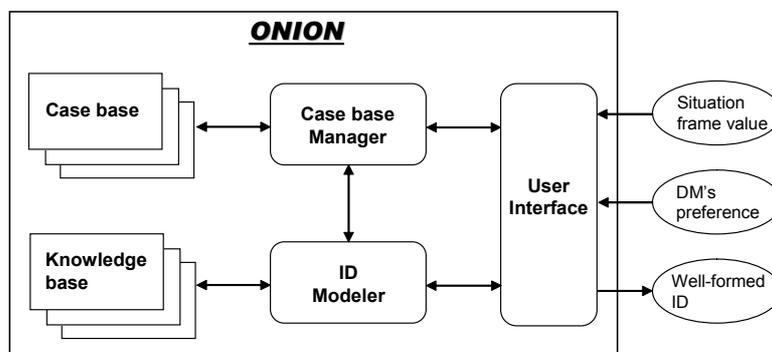


Figure 8. System architecture

User Interface module provides interactive functions for question and answer. It takes the information of current problem situation from DM. It presents combined ID of super ID and core ID, and supplement nodes to DM. The DM's decision for adaptation procedure is asked. And it shows the resulting ID of the decision problem. The interaction between the system and DM is performed through formed layout and diagram at User Interface. The Case base Manager performs following works: it retrieves the candidate cases by the case retrieval procedure presented in section 3.3 and stores the resulting case as a new case. The ID Modeler generates a super ID and a core ID by the combination of the retrieved candidate IDs, modifies the combined ID, and builds a well-formed ID using decision analytic knowledge. The decision analytic knowledge is stored at Knowledge base, which is used to evaluate whether an ID is well-formed ID or not. Case base includes the cases. A case consists of a situation and its corresponding ID of a specific decision problem, which are represented using frame type as described in previous section.

Validation

ONION is evaluated and discussed on several points: process, the number of cases, and knowledge acquisition.

Process: ONION provides the entire decision modeling process for the decision participants. The system can support for decision participants input the relevant knowledge and information, and suggest an ID based on the input and existing case base. And the system helps for the decision participants modify the suggested ID to become WFID. *The number of cases*: ONION is developed to implement CBR-based methodology for DCA. Contrast to other approaches, like neural-networks, ONION suggests a better result even if the number of cases is not enough. *Knowledge acquisition*: the decision class of the ONION can be efficiently expanded by adding the problem and its resulting ID to case base as a new case. Existing approaches such as rule-base systems and neural networks suffers difficulty in knowledge acquisition process.

CONCLUSIONS

Modeling real decision problem is a difficult, time consuming and a hard working process, involving many decision participants. ONION is developed to facilitate a decision model building process using CBR. ONION consists of the following functions: the retrieval of candidate cases, adaptation them to the specific situation of a given problem and storing the result as a new case.

Evaluation showed that our suggested CBR-based methodology and ONION have good points for implementing DCA. This research depends on decision participants to retrieve domain specific knowledge in the adaptation process. Storing domain specific knowledge at a knowledge base to lessen the burden of decision participants. Acquiring situation-specific knowledge and modification of retrieved candidate IDs are done by communication among decision participants. Therefore it is left at a further research area to suggest a group decision procedure.

APPENDIX 1. CASE RETRIEVAL PROCEDURE

1. Notation

For the more formal explanation, some notations are defined as follows:

N : the number of cases.

X_1, X_2, \dots, X_N : cases

$T = \{S_1, S_2, \dots, S_N\}$: the set of the situations, where S_k is the situation frame of X_k , $k=1, \dots,$

N .

m : the number of situation frames of each case.

$S_k=(e_{k1}, e_{k2}, \dots, e_{km})$: the value of situation frames of case $X_k, k=1, \dots, N$.

$S_0=(e_{01}, e_{02}, \dots, e_{0m})$: the value of situation frames of a current decision problem,

$R=(r_1, r_2, \dots, r_m)$: R is an indicator representing whether each situation frame of a current decision problem are covered or not. $r_j=1$ if e_{0j} is not complete covered, else then r_j has 0.

$W=(w_1, w_2, \dots, w_m)$: the weight of situation frame, which have a value ($0 < w_2 < 1$).

Let $A=(a_1, a_2, \dots, a_m)$ and $B=(b_1, b_2, \dots, b_m)$ to be situations, and a_j and b_j to be the value of j th situation frame of A and B , respectively.

$A \circ B=(c_1, c_2, \dots, c_m)$, where $c_j=(1 - |a_j - b_j|)$.

$A \otimes B=(c_1, c_2, \dots, c_m)$, where $c_j=(a_j \times b_j)$.

$A \oplus B=(c_1, c_2, \dots, c_m)$, where $c_j=(a_j + b_j)$.

$A \Delta B=(c_1, c_2, \dots, c_m)$, where $c_j=0$ if $(a_j - b_j) \leq 0$, else then $c_j=(a_j - b_j)$.

$Sum(A)=\alpha$, where $\alpha=\sum_{i=1}^m a_i$.

2. Definitions

Fitting ratio, F_k means the degree that the current situation frames of a current problem are to be covered when X_k is selected as a next retrieved case. F_k is $F_k = \frac{Sum((R \otimes W) \otimes (S_0 \circ S_k))}{Sum(R \otimes W)}$.

Garbage ratio, G_k means how many fractions of situation frames of X_k is not matched with those of a current problem. G_k is $G_k = 1 - Sum(W \otimes (S_0 \circ S_k))$.

Coverage, C_i means how much situation frames of the current problem is covered by candidate cases at stage i . C_i is $C_i = 1 - \frac{Sum(R_i)}{m}$ where $R_i = R_{i-1} \Delta (S_0 \circ S_k)$.

3. Example of using fitting ratio and garbage ratio

This section shows a numeric example of above notations and definitions. Figure 9 shows a simple example for calculating of fitting ratio and garbage ratio. The fitting ratio is calculated between the situation frames of a given problem and those of the case which have same decision nodes in a class. For example let the current situation of a current problem $S_0=(1, 1, 0, 0, 0, 0, 1)$ and existing case X_1 is $S_1=(1, 0, 0, 1, 0, 1, 0)$ then $(S_0 \circ S_1) = (1, 0, 1, 0, 1, 0, 0)$. And let's assume that four situation frames of current situation are covered until now, i.e., $R=(0, 0, 0, 1, 1, 0, 1)$ and C_1 is $4/7$. Furthermore, let's assume that the weights of current situation frames are equal then $W \otimes R = (1/7, 1/7, 1/7, 1/7, 1/7, 1/7, 1/7) \otimes (0, 0, 0, 1, 1, 0, 1) = (0, 0, 0, 1/7, 1/7, 0, 1/7)$. As $(R \otimes W) \otimes (S_0 \circ S_1) = (0, 0, 0, 1/7, 1/7, 0, 1/7) \otimes (1, 0, 1, 0, 1, 0, 0) = (0, 0, 0, 0, 1/7, 0, 0)$ and $Sum(R \otimes W) = 3/7$, so $F_1 = \frac{Sum((R \otimes W) \otimes (S_0 \circ S_1))}{Sum(R \otimes W)} = (1/7)/(3/7) = 1/3$.

As $W \otimes (S_0 \circ S_1) = (1/7, 0, 1/7, 0, 1/7, 0, 0)$, $G_k = 1 - Sum(W \otimes (S_0 \circ S_k)) = 4/7$. The fact that F_1 is $1/3 = 0.333$ implies that X_1 covers 33.3% of the uncovered situation frames of a current problem. In contrast, G_1 is $4/7 = 0.571$ implies that if X_1 is retrieved, 57.1% of situation frames of X_1 and that of current problem are incongruent.

4. Procedure

Let's define a function $MAX(T)$, which returns the situation number that has the largest value

of the ratio of fitting ratio to garbage ratio, i.e., F_k/G_k , where T is set of situations and $k=1, \dots, N$. The procedure of selecting candidate cases is summarized as follows.

A retrieval procedure

Given $T=\{X_1, X_2, \dots, X_N\}$
 $S_0=(e_{01}, e_{02}, \dots, e_{0m})$
 $S_k=(e_{k1}, e_{k2}, \dots, e_{km})$.
 $R=(r_1, r_2, \dots, r_m)$
 $S_0 \circ S_k=(a_1, a_2, \dots, a_m)$.

Step(0) Set $r_j=1, j=1, \dots, m$.

Step(1) Calculate F_k, G_k . If $F_k=0$ then goto *Step(3)*. Otherwise calculate $F_k/G_k, k=1, \dots, N$.

Step(2) Set $k = MAX(T)$.

Step(3) Select k th case.

Step(4) Set $R = R \Delta (S_0 \circ S_k)$ and calculate C_i .

Step(5) If $C_i \geq \theta_C$ then stop. Otherwise set $T = T - \{X_k\}$.

Step(6) If $T = \emptyset$ then stop. Otherwise goto *Step(1)*.

Figure 9 describes an example of selecting candidate cases. In this example, S_1, S_2 , and S_3 are situations of X_1, X_2 , and X_3 which are stored in the case base. Figure 9 shows the values of R, C_i, T, F_k, G_k and F_k/G_k at each stages.

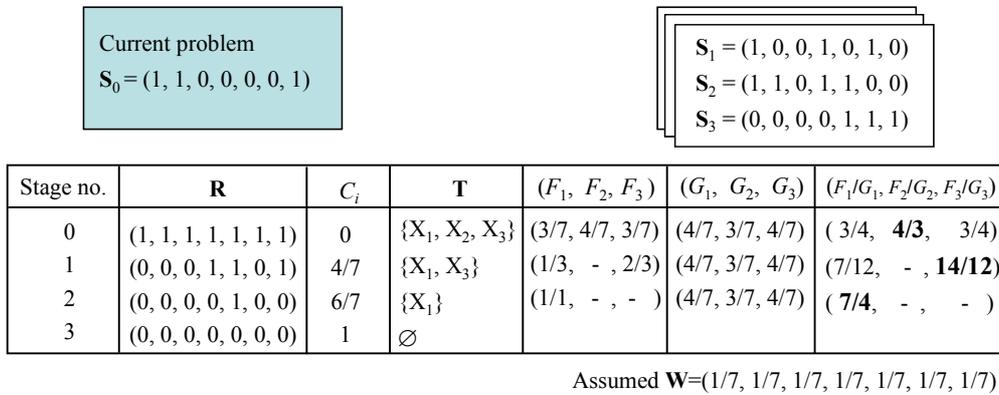


Figure 9. An example for candidate case selection

APPENDIX 2. DEFINITIONS

To testify that a constructed ID meets the conditions of well-formed ID, we suggest using the following definitions.

Definition: An *Influence diagram (ID)*, ID is a directed acyclic digraph $G = (N, A)$, where N is a finite set of nodes and A is a set of arcs, $A \subseteq N \times N$.

Definition: The *nodes*, the nodes are partitioned into three sets C, D and V . The chance node $c \in C$, which is circular shape, represents uncertain or certain states and the rectangular-shaped decision node $d \in D$ reveals variables whose values are chosen by the DM. The diamond-shaped value node $v \in V$ represents the objective to be maximized in expectation by the DM.

Definition: A *Well-formed influence diagram (WFID)*, WFID is an ID, (1) the directed graph has no cycles, (2) at least one more decision nodes and chance nodes, (3) the value node, if present, has no successors, and (4) there is a directed path that contains all of the decision nodes. For the further information about this, please refer Shachter (1986).

REFERENCES

- Anandanpillai, T., T. Barta, A case-based reasoning system for housing discrimination law, *Expert Systems With Applications*, 16, 1999, 315-324.
- Alexander, P., G. Minden, C. Tsatsoulis, J. Holtzman, Storing design knowledge in cases, *Proceedings of the Workshop Case-Based Reasoning(Pensacola Beach FL)*, 1989.
- Ashley, K., *Modeling Legal Argument: Reasoning with Cases and Hypothetical*, MIT Press, Cambridge, MA, 1990.
- Branting, L., *Integrating Rules and Precedents for Classification and Explanation: Automating Legal Analysis*, PhD thesis, Department of Computer Science, University of Texas, Austin, USA, 1991.
- Chiu, C., A case-based customer classification approach for direct marketing, *Expert Systems With Applications*, 22, 2002, 163-168.
- Chung, T.Y., J.K. Kim, S.H. Kim, Building an influence diagram in a knowledge-based decision system, *Expert Systems With Applications* 4, 1992, 33-44.
- Cooper, G.F., E. Herskovits, A Bayesian Method for Constructing Bayesian Belief Networks from Databases, *Proceeding of the Seventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufman Publisher, 1991, 86-94.
- Dayal, S., M. Harmer, P. Johnson, D. Mead, Beyond knowledge representation: commercial uses for legal knowledge bases, *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, ACM, New York, NY, 1993.
- Egar, J.W., M.A. Musen, Graph-Grammar Assistance for Automated Generation of Influence Diagram, *IEEE Transaction on Systems, Man, and Cybernetics*, 24, 1994, 1625-1642.
- Fdez-Riverola, F., J.M. Corchado, CBR based system for forecasting red tides, *Knowledge-Based Systems*, 16, 2003, 321-328.
- Grupe, F.H., R. Urwiler, N.K. Ramarapu, M. Owrang, The application of case-based reasoning to the software development process, *Information and Software Technology*, 40, 1998, 493-499.
- Hammond, K., *Case-Based Planning: Viewing Planning as a Memory Task*(New York: Academic), 1989.
- Herkman, D., E. Horvitz, Problem Formulation as the Reduction of Decision Model, *Proceeding of the Seventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufman Publisher, 1991, pp. 159-184.
- Holtzman, S., *Intelligent Decision Systems*(Addison-Wesley MA), 1989.
- Howard, R.A., The used car buyer in R.A. Howard, J.E. Matheson (Eds.), *Readings on the Principles and Applications of Decision Analysis Vol II*(Strategic Decision Group, Menlo Park, CA), 1984.
- Howard, R.A., Decision analysis: practice and promise, *Management Science* 34, 1988, 679-695.
- Jarmulaka, J., E. J.H. Kerckhoffs, P. Veen, Case-based reasoning for interpretation of data from non-destructive testing, *Engineering Applications of Artificial Intelligence*, 14, 2001, 401-417.
- Kervahut, T., J. Potvin, An interactive-graphic environment for automatic generation of decision tree, *Decision Support Systems*, 18, 1996, 117-134

Kim, J.K., A Knowledge-Based System for Decision Analysis. Ph.D. thesis, Department of Industrial Engineering(KAIST Korea), 1991.

Kim, J.K., Knowledge Acquisition for Knowledge-based Decision Systems, *Applied Artificial Intelligence* 11, 1997, 131-149.

Kim, J.K., K.S. Park, Neural network-based decision class analysis for building topological-level influence diagram, *Int. J. Human-Computer Studies* 46, 1997.

Kim, J.K., S.C. Chu, Analyzing a Class of Decision Problems: Neural Network Based Approach, *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, 1998, 36-43.

Kim, J.K., J.K. Lee, S.H. Lee, An Interactive Approach to Build an Influence Diagram Based on Neural Networks, *Journal of Decision Systems*, 8, 1999, 1-17.

Kolodner, J., Improving Human Decision Making through Case-Based Decision Aiding(AI MAGAZINE, Summer), 1991.

Kolodner, J., *Case-Based Reasoning*(Morgan Kaufmann Publishers, CA), 1993.

Lee, J.K., J.K. Kim, S.H. Kim, A Methodology for Modeling Influence Diagrams: A Case-Based Reasoning Approach, *International Journal of Intelligent Systems in Accounting, Finance and Management*, 9(1), 2000, 55-63

Luua, D.T., S.T. Ngb, S.E. Chen, A case-based procurement advisory system for construction, *Advances in Engineering Software*, 34, 2003, 429-438.

Machado, A.M.C., M.F.M. Campos, A Connectionist Approach for Building Influence Diagrams, *Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences*, 1997, 68-73

Olmsted, S.M., On Representing and Solving Decision Problems. Ph.D. thesis, Department of Engineering-Economic Systems(Stanford University), 1984.

Pal, K., O. Palmer, A decision support system for business acquisitions, *Decision Support Systems*, 27, 2000, 411-429.

Porter, B.W., R. Bareiss, R.C. Holte, Concept learning and heuristic classification in weak-theory domain, *Artificial Intelligence* 45, 1990, 229-263.

Reategui, E.B., J.A. Campbell, B.F. Leao, Combining a neural network with case-based reasoning in a diagnostic system, *Artificial Intelligence in Medicine*, 9, 1997, 5-27.

Reed, J., Building decision models that modify decision systems, *Knowledge System Laboratory No. KSL-89-21*(Stanford University, Stanford, CA), 1989.

Risebeck, C.K., R.S. Schank, *Inside case-based reasoning*(Northvale, NJ: Erlbaum), 1989

Sarkar, S., R. S. Sriram, S. Joykuty, I. Murthy, An information theoretic technique to design belief network based expert systems, *Decision Support Systems* 17, 1996, pp.13-30.

Shachter, R.D., Evaluating Influence Diagrams, *Operations Research* 34, 1986, 871-882.

Sonnenberg, F.A., C.G. Hagerty, C.A. Kulikowski, An Architecture for Knowledge-based Construction of Decision Models, *Medical Decision Making* 14, 1994, 27-39.

Suh, M.S., W. C. Jhee, Y.K. Ko, A. Lee, A case-based expert system approach for quality design, *Expert Systems With Applications*, 15, 1998, 181-190.

Tessem, B., S. Bjornestad, Analogy and Complex Software Modeling, *Computers in Human Behavior*, 12(4)

Wen, W.X., From Relational Databases to Belief Networks, *Proceeding of the Seventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufman Publisher, 1991, pp. 406-413.