Int J Adv Manuf Technol (2004) 23: 532–540
DOI 10.1007/s00170-003-1676-0

ORIGINAL ARTICLE

S. L. Wang · S. H. Hsu

A Web-based CBR knowledge management system for PC troubleshooting

Received: 11 November 2002 / Accepted: 13 February 2003 / Published online: 12 February 2004

Abstract Using case-based reasoning (CBR), the authors integrate the techniques of cognitive task analysis (CTA), hierarchical clustering and ontology and propose a Web-based CBR knowledge management (KM) system for investigating the construction of a KM system with multiple information techniques to support KM activity in industry. The maintenance service centres of a computer company are used as an example to illustrate extracting the maintenance knowledge necessary to construct a PC troubleshooting KM system. The effectiveness of applying a Web-based CBR KM system to support KM activities in the KM life cycle is subjected to practical verification.

Keywords Case-Based Reasoning (CBR) · KM System · Cognitive task analysis (CTA) · Ontology · Hierarchical Clustering

1 Introduction

The concept of knowledge management (KM) was pointed out in the early 1990s. Only recently, however, has it received attention in the practical industrial domain primarily because KM and innovative knowledge are becoming too important for industries to ignore in facing global competition. Organizations, according to Grundstein and Barthès [1], are made up not only of their products and service units, but also of their knowledge assets. It is therefore necessary for industrial units to build up KM systems appropriate to their scales and requirements. Such knowledge systems can provide benefits in the following ways:

S. L. Wang · S. H. Hsu (✉)
Institute of Industrial Engineering,
National Chiao Tung University,
Hsinchu, Taiwan ROC
E-mail: shhsu@cc.nctu.edu.tw
Tel.: +886-3-5726731
Fax: +886-3-5722392

– preventing the loss of know-how when professionals leave the organization;
– taking advantage of knowledge and techniques previously gained from experience so as not to re-make mistakes;
– developing organizational knowledge maps that can serve as guidelines in making manufacturing strategies;
– helping with information cycling and communication among various units;
– enhancing employee learning environments;
– integrating know-how from various sources in organizations.

Nowadays, many manufacturers are facing serious structural problems brought about by the rapid development of overseas activities such as factories [2], branch companies and manufacturing facilities set up in various areas to meet business expansion requirements. Facilities located in different regions greatly split core knowledge and make it more difficult to carry out KM activities. It is therefore worthwhile to conduct an in-depth investigation into how divergent industrial knowledge can be systematically integrated so as to obtain effective KM. The rapid development of information-handling techniques over the past decade has made knowledge-based systems, including expert systems, corporate memory systems, information systems and other advanced information resources indispensable to organizations seeking effective KM [3].

Though there are many researchers dedicating themselves to the development of KM techniques, there is currently no single information system that supports all the activities in the KM cycle. Typically, many individual information systems supporting various KM activities are offered. Many such KM systems put considerable emphasis on the knowledge storage and memory aspects. However, Kendall [4] pointed out that it is necessary to integrate related information techniques in the development of KM systems to guarantee that all activities in the KM cycle are sufficiently supported.

To ensure the effective integration of information techniques, the authors combined CTA, ontology techniques and a Web-based case-based reasoning (CBR) model to develop a KM system. Multiple techniques were incorporated to support industrial KM activities, including knowledge capture, knowledge development, knowledge sharing and knowledge utilization. Moreover, a computer company was used as a practical case to investigate extracting the maintenance know-how required for PC troubleshooting and diagnosis. The effectiveness of using Web-based CBR to enhance KM activities was also assessed.

2 System architecture

The authors applied CBR to develop a Web-based CBR model and finally a KM system, and an approach that has the following advantages:

- Experience knowledge scattered among various sites in different areas can be integrated in a unified format.
- CBR allows the continuous updating and adaptation of corporate memory. This refines and enriches knowledge library content, and builds up the KM system.
- The more problems the CBR system solves, the wider the scope of problems it can cover. Recycling of experience knowledge in the same problem domains will reduce the incidence of trial-and-error.
- Because CBR resembles human reasoning, the problem-solving ability of an organization's professional personnel is upgraded with CBR support.
- Using CBR as a real-time Internet consultant can enhance knowledge communication and sharing among employees, as well as the organizational learning environment.

2.1 The structure of the Web-based CBR model

Schank [5] and Riesbeck and Schank [6] advocated CBR and referred to it as an alternative to traditional rule-based and model-based reasoning. In recent years, CBR techniques have been applied to a wider range of problem domains including catering, recipe-making, dispute mediation, criminal sentencing and process planning [7, 8, 9]. Various computer-assisted systems have been developed for industrial tasks such as in injection moulding and design [10], architecture design [11], fixture design [12], process planning [13] and die-casting die design system [14].

According to Sengupta, Wilson and Leake [15], we can identify three CBR implementation models: task-based, enterprise-based and Web-based. The task-based model is a traditional CBR system designed for a specific task and doesn't include knowledge-sharing functions. The enterprise model is a system constructed for an enterprise to manage proprietary knowledge such as project experience, problem-solving methods, etc. As its name suggests, the Web-based model breaks geographical barriers through the World Wide Web (WWW), thus making it possible for scattered enterprise units to share knowledge.

On account of its characteristics, a Web-based CBR model was employed in the present study to help build up the CBR distribution system. Intelligent Web-based case assistants were designed using a thin-client structure. Communication between client and server, as well as the user interface was implemented at the client end. All the business logic and the logic integrating the two ends was confined to the server end. The structure of the Web-based CBR model is shown in Fig. 1. The proposed architecture is shown in Fig. 2.

2.2 The CBRKM system structure

The KM system proposed in this study consists of the following components:

- A user interface: for queries and case knowledge acquisition
- A KM Module: this module is mainly for acquiring case-related knowledge so as to build up and maintain databases such as the case library, the ontology library, the similarity matrix library and the global vocabulary library.
- A CBR engine: When users input case attributes, this processes the computing algorithm and prompts with similar cases for reference.
- A case knowledge sharing converter: the major function of this module is to offer standards for the translation and the mapping of domain knowledge elements.

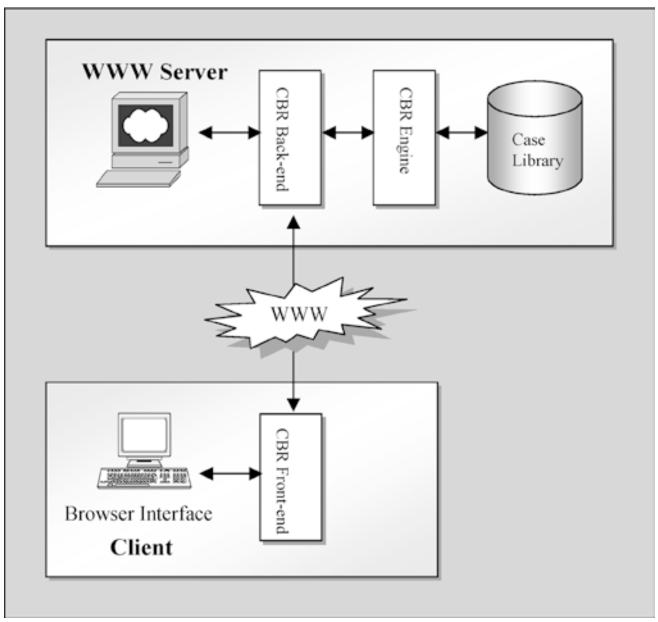

Fig. 1 The structure of the Web-based CBR model

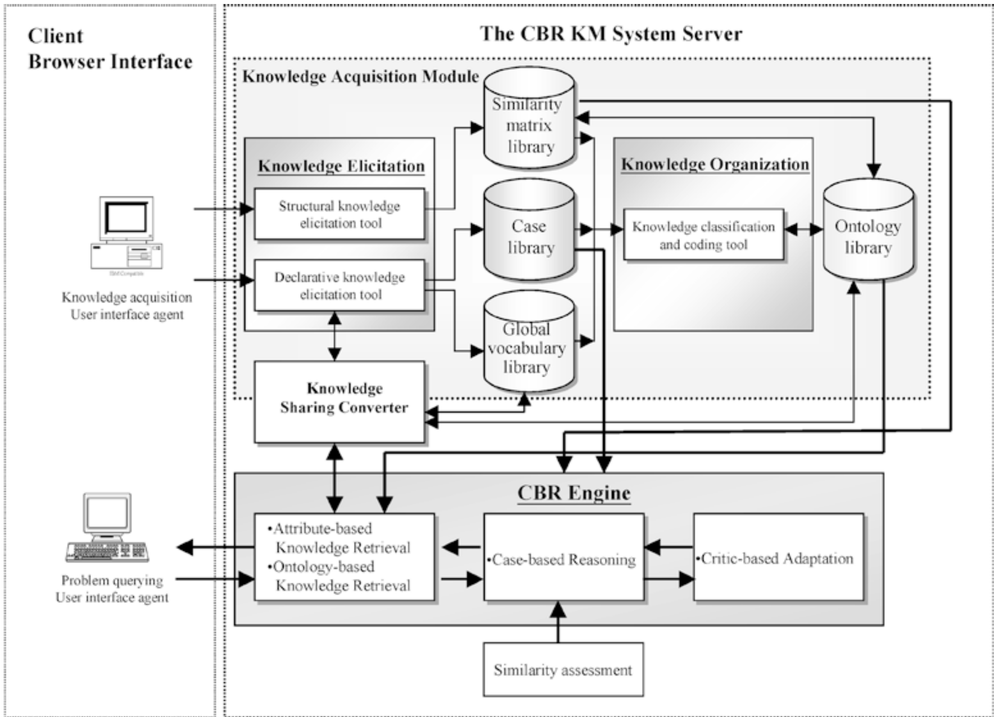

Fig. 2 The proposed architecture of the Web-based CBRKM system

– A case library: case knowledge stored in the case library includes case attribute indices, a declarative vocabulary and problem solution knowledge.

KM activities during the KM cycle include knowledge capture, knowledge development, knowledge sharing and knowledge utilization, and each of the system components plays a role in those activities. The KA module and the case library handle knowledge acquisition support. The CBR engine and knowledge classification and coding tools handle organization and development, enabling case knowledge to be translated into suitable formats for sharing and retrieval. The case knowledge sharing converter changes terminology in different units into standard vocabulary so that knowledge from different sources can be communicated and shared. More importantly, adoption of the Web-based CBR model helps dissemination of knowledge. Above all, use of an ontology-based interface enhances the reuse and sharing of knowledge. The relationship between the functional module and the KM activity in KM life cycle in the CBRKM system is illustrated in Fig. 3.

3 Implementation approach

In the study, the Java programming language and a dynamic server Web page were used to construct the Web-based CBRKM system because they are platform-independent, Internet-supported and suitable for developing KM systems for the Internet. The CBR reasoning mechanism and ontology techniques were employed in constructing the case knowledge KM system. Related methods and processes are described in the following section.

3.1 The Case retrieval algorithm

The RETRIEVE process deals with the case similarity measure, which compares query cases and old cases to find the cases most likely to be useful, while the case indexing procedure provides an efficient way to search for candidates. Users need not understand the relationship between the description and solution parts since an automatic reasoning algorithm feeds back proposed solutions.

The case retrieval algorithm (Eq. 1) described in the study was mainly derived from an algorithm proposed by

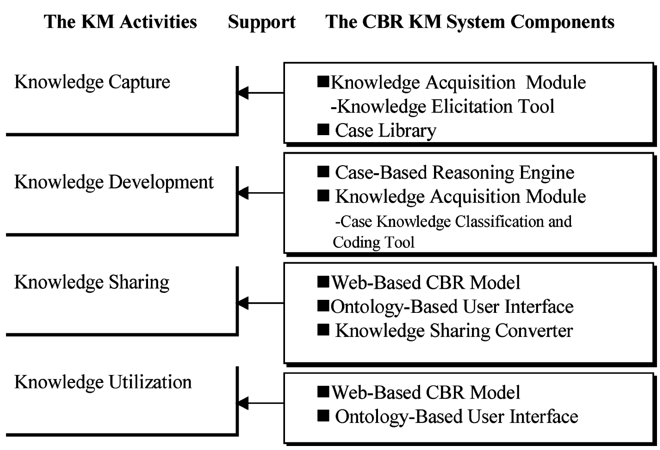

Fig. 3 The relationship between the system components and KM activity in the CBRKM system

Janet Kolodner [16] that determines similarities between cases and identifies those with higher similarity values.

$$\frac{\sum_{i=1}^{n} W_i \times Sim\left(f_i^I, f_i^R\right)}{\sum_{i=1}^{n} W_i} \tag{1}$$

$n =$	the number of attribute indexes;
$W_i =$	the weighting value of each attribute index;
$f_i^I = f^I{}_i =$	newly entered case;
$f_i^R =$	case in the case library;
$Sim\left(f_i^I, f_i^R\right) =$	the similarity between the entered case and the case in the case library.

3.2 The case knowledge organization approach

Ontology is a collection of key concepts and their interrelationships that collectively provide an abstract view of an application domain [17, 18]. With the support of ontology, users can communicate with one another and with the system with a shared and common understanding of domain knowledge. In the study, the algorithm proposed by Uschold and Grüninger [18] was used as a foundation for developing the case knowledge organization approach. Practically speaking, the organisation of case knowledge can be categorized into five steps: data preprocessing, structuralizing case knowledge, building up the domain concept hierarchy, formalising ontology and evaluation. Details on how case knowledge is organized are given below:

3.2.1 Step1—Data preprocessing

(1) Capture indices and descriptive case attribute vocabularies. Important case attribute indices as well as their vocabularies are extracted during this stage.
(2) Rate case attributes to determine attribute similarity values. The degrees of similarity between cases are then computed according the CBR algorithm.

3.2.2 Step2—Structuralizing case knowledge

(1) Categorize cases put in preliminary groups by hierarchical clustering [19]. Using hierarchical clustering allows homogeneous cases to be grouped according to their clustering threshold values, and also allows for processing noisy data.
(2) Evaluate case clustering results. The evaluation indices proposed by Hsu et al. [20] are used for the evaluation task to ensure the rationality and consistency of clustering results. These indices determine reasonable numbers of clusters and suitable clustering threshold values that serve as judgment criteria for automatic classification of new cases.

3.2.3 Step 3—Building up the domain concept hierarchy

A domain concept hierarchy is needed to represent the ontology structure. In this study the domain concept hierarchy is defined using the object-oriented approach. A domain ontology defines roles and their relationships. A role represents a real-world concept, while relationships between roles are defined through two relations: IS_A and HAS_A. The IS_A relation shows subclass and inheritance. For example, "X IS_A Y" indicates X is a sub-class of Y and inherits the attributes of Y, and Y is a super-class of X. HAS_A relation represents a part-whole relation. For example, "X HAS_A Y" indicates Y is an element of X. A defined domain concept hierarchy forms a classification structure with varying degrees of abstractness and concreteness. In such a structure, concepts on the abstract level usually provide less detailed information than those on the concrete level. Moreover, many concrete concepts may share abstract concepts. These principles and specifications allow such a domain concept hierarchy to be applied to practical domains.

3.2.4 Step 4—Formalizing the case knowledge ontology

A semantic hierarchical structure for groups of cases is constructed based on the domain concept hierarchy in this step. The structure starts with the lowest concept level and gradually goes to higher levels until the semantic structures of all groups are organized. The combination of group semantic structures completes the ontology of the application domain.

3.2.5 Step 5—Evaluation

To evaluate the case knowledge ontology, the present software environment and documents will be used to assess how to build up the ontology through programming languages.

4 Case study—PC troubleshooting

The knowledge pattern in an industry can be know-how, maintenance facts, product requirements, design rationale, experience or professional knowledge. Among them, know-how is an important element in that it contains problem-solving expertise in functional disciplines, experience of human resources, process experience, design issues and lessons learned. However, such knowledge must be accumulated through systematic acquisition and storage. It is, therefore, a fundamental job for industries to systematically integrate dispersed know-how when building up KM systems. In this study, the authors used the PC troubleshooting maintenance centres of a computer company as an example of applying a systematic method for extracting troubleshooting know-how and maintenance facts to build up a PC troubleshooting KM system.

Knowledge elements in this study are described according to case patterns. In general, they consist of declarative case knowledge and structural case knowledge. Declarative case knowledge consists of two parts: a description part and a solution part. The description part describes case attributes via certain indexes. The solution part is the major case knowledge component, contains the know-how. Structural case knowledge contains similarities among case attributes.

4.1 Troubleshooting knowledge acquisition

The methods for case knowledge acquisition were as follows.

(1) Structured interviews and concept elicitation methods [21] were applied to acquire declarative case knowledge, including possible attribute features, attribute indices and descriptive troubleshooting vocabulary.
(2) Structured interviews and cognitive task analysis methods [21] were used to acquire the subjects' problem-solving knowledge. The authors applied the GOMS (Goal, operators, methods, and selection rules) developed by Card et al. [22] in which a series of open-ended questions are used to lead subjects to verbally report on troubleshooting processes.
(3) The authors applied rating tasks to evaluate case attribute similarities. Three case-attribute similarity matrices for the fault attribute indices were obtained, which enabled calculation of further case similarity matrices.

4.2 Constructing the hierarchical case knowledge classification structure

A hierarchical case knowledge classification structure for PC troubleshooting was built up based on the case knowledge organization approach described in Sect. 3.2. The algorithm for doing so consists of the following steps.

(1) Extract case attribute index and declarative vocabulary. Important PC-troubleshooting case attributes such as fault condition, fault position and fault symptom were first extracted and a declarative vocabulary for these attribute indices identified. For example, fault condition vocabulary might include "no display", "system failed to start" and "failed to connect to the Internet" etc., fault position vocabulary might include "CPU, power supply", "hard drive" etc., and fault symptom vocabulary might include "CMOS RAM error", "games and programs run too fast", "Windows protection error", etc.
(2) Calculate case similarities. After the declarative case attribute vocabulary was extracted, attribute

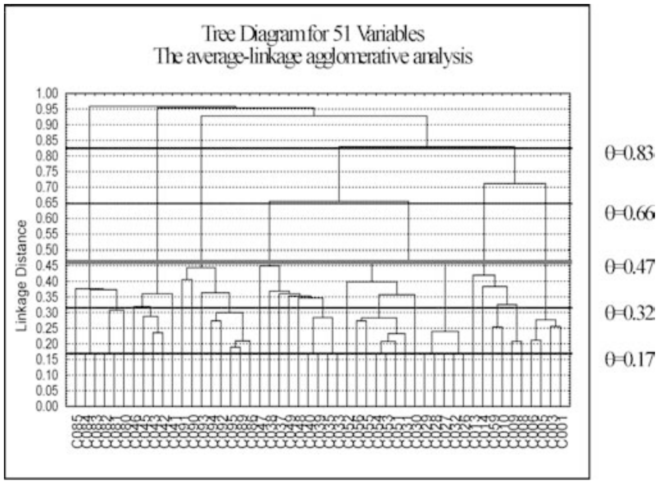

Fig. 4 Hierarchical clustering outcomes

weighting values for troubleshooting were determined by maintenance experts to be: fault condition 33%, fault location 15% and fault symptom 52%. The experts were also asked to rate the similarities between case attributes so as to build up a matrix of case attribute similarities. The matrix of case similarities was then computed using the CBR algorithm.
(3) Structuralize case knowledge. An average-linkage agglomerative analysis of the hierarchical clustering was first conducted and the output can be seen in Fig. 4. The clustering rationality and consistency indexes proposed by Hsu et al. [20] were used to assess the consistency and rationality of clustering results derived from computed clustering threshold values and the sorting done by the experts. A suitable clustering threshold value was determined by evaluating a data set of 51 training items. The results are shown in Table 1. When the clustering threshold value θ was 0.47, the rationality index of the clustering output was 0.857, and the consistency index 0.902. These indices were the highest among the clustering threshold values. Through hierarchical clustering analysis, cases were divided into seven groups and the clustering threshold value criterion for automatic classification of new cases was set at 0.47.
(4) Build up the domain concept hierarchy. The object-oriented approach was used to build up the PC troubleshooting domain conceptual hierarchy. In the classification structure, fault condition was the first level. The second level is the fault position, meaning the positions that are disabled. The third level, fault symptom, represents specific problem indications and error messages emitted by the computer. The lowest levels are the most concrete and detailed information in the hierarchy.
(5) Formalise ontology. At this stage, coding of the case knowledge is translated into concrete form.

Table 1 A comparison of expert classifications and various clustering in different threshold values

Classification type	Expert Classification	Hierarchical Clustering in different threshold values				
		θ=0.83	θ=0.66	θ=0.47	θ=0.32	θ=0.17
Case Clustering	[C001 C003 C005 C006] [C008 C009 C010 C013 C014 C059] [C029 C028 C027 C032 C026] [C030 C031 C051 C052 C053 C054 C055 C056] [C047 C038 C037 C049 C048 C040 C039 C035 C033] [C091 C090 C093 C094 C092 C095 C089 C086] [C046 C045 C043 C042 C041] [C085 C084 C083 C082 C081 C080]	[C006 C005 C003 C001 C013 C014 C059 C010 C009 C008 C029 C028 C027 C032 C026 C052 C056 C055 C054 C053 C051 C031 C030 C047 C038 C037 C049 C048 C040 C039 C035 C033] [C091 C090 C093 C094 C092 C095 C089 C086] [C046 C045 C043 C042 C041] [C085 C084 C083 C082 C081 C080]	[C006 C005 C003 C001] [C013 C014 C059 C010 C009 C008] [C029 C028 C027 C032 C026 C052 C056 C055 C054 C053 C051 C031 C030 C047 C038 C037 C049 C048 C040 C039 C035 C033] [C091 C090 C093 C094 C092 C095 C089 C086] [C046 C045 C043 C042 C041] [C085 C084 C083 C082 C081 C080]	[C006 C005 C003 C001] [C013 C014 C059 C010 C009 C008] [C029 C028 C027 C032 C026 C052 C056 C055 C054 C053 C051 C031 C030] [C047 C038 C037 C049 C048 C040 C039 C035 C033] [C091 C090 C093 C094 C092 C095 C089 C086] [C046 C045 C043 C042 C041] [C085 C084 C083 C082 C081 C080]	[C006 C005 C003 C001] C013 C014 [C059 C010] [C009 C008] [C029 C028 C027 C032 C026] C052 [C056 C055 C054 C053 C051 C031] C030 C047 C038 C037 C049 C048 [C040 C039 C035 C033] C091 C090 [C093 C094 C092 C095 C089 C086] [C046 C045 C043 C042] C041 C085 [C084 C083] [C082 C081 C080]	C006 C005 C003 C001 C013 C014 C059 C010 C009 C008 [C029 C028 C027] [C032 C026] C052 C056 C055 [C054 C053] C051 C031 C030 C047 C038 C037 C049 C048 [C040 C039] [C035 C033] C091 C090 C093 C094 C092 C095 C089 C086 C046 C045 C043 C042 C041 C085 [C084 C083] [C082 C081] C080
Number of Groups	8	4	5	7	23	43
Appropriate number of group (r)		0.429	0.571	0.857	0.651	0.186
The level of homogeneity (h)		28/51=0.549	38/51=0.745	46/51=0.902	34/51=0.667	13/51=0.254

According to the domain concept hierarchy, the lowest levels of the seven groups were first examined for semantic structures, which were then combined to form the hierarchical case knowledge classification structure. From the hierarchical case knowledge classification structure, the users can connect to the troubleshooting knowledge in the case library. Finally, the authors integrated the opinions of maintenance experts, and the ontology of PC trouble-shooting was built up.

4.3 System functions and operation

A PC-troubleshooting CBR KM system was developed on the basis of the Web-based CBRKM system architecture. System functions and operation are described below.

– User Interface: An ontology-based user interface encoded in programming language was developed. With such a graphical interface, users can search for and retrieve needed case knowledge. The user interface for case knowledge retrieval is shown in Fig. 5.
– KA Module: The purpose of this module was to retrieve related domain knowledge so as to build and maintain the case library. KA tools provided in the system include the case attribute rating tool, the declarative case knowledge retrieval tool and the automatic case knowledge classification and coding tool.
– CBR Engine: When users select fault attributes via the user interface, the system automatically processes the reasoning algorithm and lists similar cases. Fig. 6 shows the output display after case reasoning.

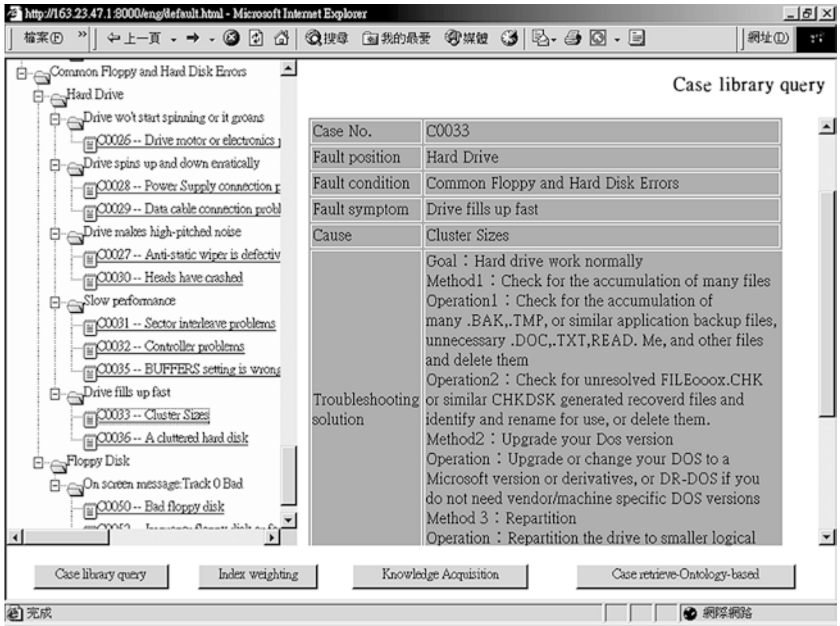

Fig. 5 The user interface for case knowledge retrieval

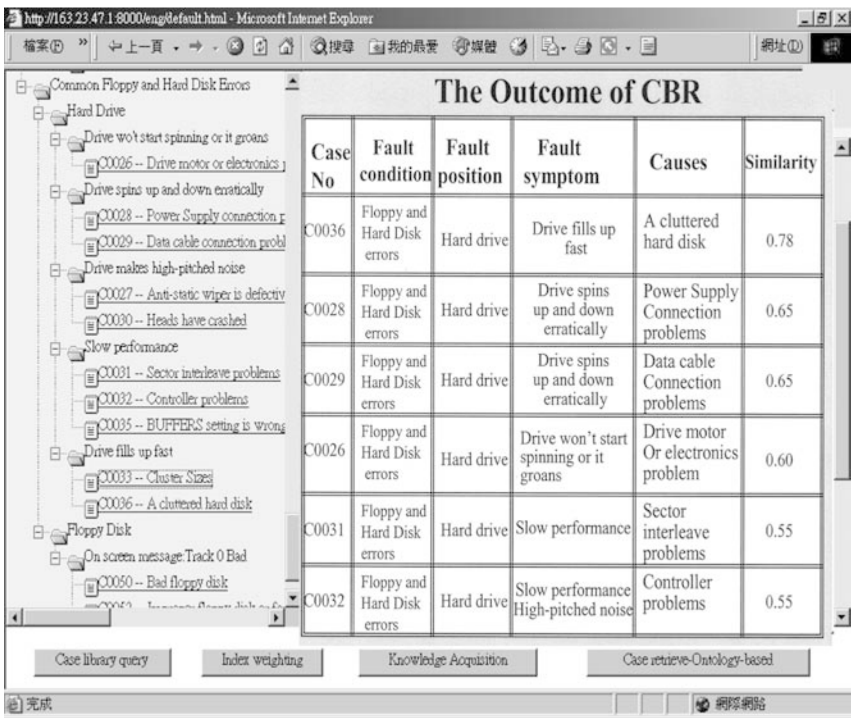

Fig. 6 The output display of case reasoning

– Case Knowledge Sharing Converter: This module provides translation and contrast standards for declarative case knowledge. The standardised interpreter makes enables various maintenance service centers to communicate and share PC troubleshooting knowledge.
– Case library: Troubleshooting case knowledge is stored in the case library where case attributes serve to index all cases. Methods and knowledge needed for PC troubleshooting are recorded for every case (Fig. 4).

5 Discussions

The effectiveness of the Web-based CBRKM system on KM activities was discussed. Primary activities in the KM cycle are described below.

(1) Knowledge Capture: Knowledge capture is the process by which knowledge is obtained and stored [23]. The KM system developed in the study adopted CTA technique to build KA tools by which engineers in maintenance service centers can directly extract PC troubleshooting knowledge. In particular, significant savings in the energy and time necessary for knowledge retrieval can be realised with the help of software. The troubleshooting knowledge extracted by maintenance engineers can then be systematically entered into the case library, making it possible for maintenance knowledge scattered among various maintenance centres to be captured and transmitted.
(2) Knowledge Development: Once knowledge has been captured, it must be organized and analyzed for strategic or tactical decision-making. Such applications are a means of gathering meaningful knowledge from existing data stored in databases, data warehouses and digital libraries [23]. Using the CBR reasoning algorithm, the system generates a matrix of similarities between cases from which a hierarchical clustering is processed to classify and structuralize the case knowledge. Moreover, the application of ontology techniques helps classify and encode the fault cases, adding the practical value of gathering PC troubleshooting knowledge dispersed and tacit in different maintenance centres. This helps maintenance engineers enhance their troubleshooting efficiency. Furthermore, the shortcomings of having such knowledge scattered and lacking in structure for reference value are also reduced.
(3) Knowledge Sharing: Once knowledge has been analyzed, distribution and sharing is the next necessary step in the process of KM. With the development of KA tools, tactic knowledge scattered among maintenance service centers can be organized and encoded for storage in the case library and ontology library. Maintenance engineers indifferent centres can communicate and share their troubleshooting experience through computer-assisted telecommunications, and the Web-based CBR model. This enhances the learning environment in the organization.
(4) Knowledge Utilization: The last step in KM is to effectively encourage employees to use knowledge. It requires vast financial resources and time commitments for organizations to build knowledge-based systems. Accordingly, information systems should be developed for end-user convenience and make it easy for users to manipulate knowledge. The authors built a hierarchical case knowledge classification structure, which was developed into a case-knowledge KM system through a graphic ontology-based user interface. With the ontology-based user interface, maintenance personnel can easily retrieve and use PC troubleshooting knowledge, thus enhancing the effectiveness of the retrieval, sharing and usage of troubleshooting knowledge.

6 Conclusions

The authors used a Web-Based CBR KM system structure to build a prototype of PC troubleshooting KM system. It has been found that organizations need to integrate different methods and techniques in developing KM systems so as to uplift the effectiveness of KM activities.

With the rapid development of industrial techniques, it is necessary for industrial organizations to realize how to hand over their experience and maintenance knowledge through a KM system so as to upgrade their innovation and development abilities. In the study, various information techniques were integrated to build an information KM system for an enterprise. The KM system construction and structure proposed in the study may serve as a guide for industrial organizations to develop KM systems.

However, it is not easy to assess the effectiveness of introducing a KM system to an organization in a short period of time. It is necessary to conduct long-term evaluations and improvements to make the KM system meet the organization's needs.

References

1. Grundstein M, Barthès JA (1996) An industrial view of the process of capitalizing knowledge. In: Schreinmakers JF (ed) Knowledge management: organization, competence and methodology. Ergon, Würzburg
2. Ong SK, An N, Nee AYC (2001) A Web-based fault diagnostic and learning system. Int J Adv Manuf Technol 18:502–511
3. Damodaran L, Olphert W (2000) Barriers and facilitators to the use of knowledge management systems. Behav Info Technol 19(6):405–415
4. Kendall KE (1997) The significance of information systems research on emerging technologies: seven information technologies that promise to improve managerial effectiveness. Dec Sci 28:775–792

5. Schank RC (1982) Dynamic memory: a theory of learning in computers and people. Cambridge University Press, Cambridge
6. Riesbeck C, Schank R (1989) Inside case-based reasoning. Lawrence Erlbaum, Hillsdale
7. Kolonder J, Simpson RL (1989) The MEDIATOR: analysis of an early case-based problem solver. Cog Sci 13:507–549
8. Hammond KJ (1989) Case-based planning: viewing planning as a memory task. Academic Press, Boston
9. Yang H, Lu WF, Li AC (1992) A framework for using case based reasoning in automated process planning. Conc Engin 59:101–114
10. Kwong CK, Smith GF (1998) A computational system for process design of injection moulding: combining a blackboard-based expert system and a case-based reasoning approach. Int J Adv Manuf Technol 14:350–357
11. Hua K, Faltings B (1993) Exploring case-based Design-CADRE. Art Intellig Eng Des Anal Manufact 7:135–144
12. Sun SH, Chen JL (1996) A fixture design system using case-based reasoning. Engin Appl Art Intellig 9(5):533–540
13. Humm B, Schulz C, Radtke M, Warnecke G (1991) A system for case-based process planning. Comp Indust 17:169–180
14. Lee KS, Luo C (2002) Application of case-based reasoning in die-casting die design. Int J Adv Manuf Technol 20(4):284–295
15. Sengupta A, Wilson DC, Leake DB (1999) On constructing the right sort of CBR implementation. In: Proceedings of the IJCAI-99 workshop on automating the construction of case based reasoners, Stockholm, Sweden, August 1999
16. Kolodner J (1993) Case-based reasoning. Morgan Kaufmann, San Mateo
17. Khan L, McLeod C (2000) Audio structuring and personalized retrieval using ontologies. IEEE Proceedings of advances in digital libraries, Washington DC, 22–24 May 2000
18. Uschold M, Grüninger M (1996) Ontologies: principles, methods and applications. Know Engin Rev 11:93–136
19. Dubes RC, Jain AK (1988) Algorithms that cluster data. Prentice-Hall, Englewood Cliffs
20. Hsu SH, Hsia TC, Wu MC (1997) A flexible classification method for evaluating the utility of automated workpiece classification system. Int J Adv Manuf Technol 13:637–648
21. Cordingley E (1989) Knowledge elicitation techniques for knowledge-based systems. In: Diaper D (ed) Knowledge elicitation. Ellis Harwood, Chichester
22. Card SK, Moran TP, Newell A (1983) The psychology of human computer interaction. Lawrence Erlbaum, Hillsdale
23. Lee SM, Hong S (2002) An enterprise-wide knowledge management system infrastructure. Ind Manage Data Syst 17–25