Using SQL in CBR for Similarity Retrieval: The Case of the TQF Advisory System

Putsadee Pornphol, Phuket Rajabhat University, Thailand
Suphamit Chittayasothorn, King Mongkut's Institute of Technology Ladkrabang, Thailand

The Asian Conference on Society, Education & Technology 2016
Official Conference Proceedings

Abstract
Case-Base Reasoning (CBR) is a methodology that stands out as one the most useful artificial intelligence techniques. The essential idea of CBR is to answer user's queries by comparing them with problems in the case base that have been solved and determines the most similar one. Case retrieval is a procedure that a retrieval algorithm finds the most similar cases to the present problem. While conventional database management systems offer restricted query flexibility, systems that can create similarity based queries, for example, those found is case-based reasoning research, would improve the utility of data resources. This paper explains a strategy for building case-based systems utilizing a conventional relational database (RDB). The similarity computing in which database queries retrieve similar cases are presented. The implementation utilizes Structured Query Language (SQL) to find such similar cases.

Keywords: Case-Based Reasoning (CBR), SQL, Similarity Retrieval, Advisory System

iafor
The International Academic Forum
www.iafor.org

1. Introduction

Case-based reasoning (CBR) is a method for solving a new problem that occurs by bringing a solving method that has been used for solving similar problem in the past to be a guideline for future problem solving. Case-based reasoning has been used for solving problems in various fields such as medicine, engineering, finance, banking, education, and tourism. Case-based reasoning is a suitable problem solving method because it is similar to human's thought process which uses past experience for solving present problem and learns from what happens [1] [2]. The important thing for Case-based reasoning is how to store old cases and a method for determining the similarity between a new case and all old cases that will give a case that is the best answer or the most similar one.

For the determination of similarity of Case-based reasoning, there are many ways such as nearest neighbor, induction, statistics, neural networks, fuzzy logic, and production rules. Case-based reasoning that uses SQL to determine the similarity of cases, store the cases in relational databases. DBMS is responsible for managing the data in that database. Then SQL statements are used to determine the similarity of the new case and the old cases. When a user of the system needs to compare a new case to find a matched old case stored in the databases, there is a problem when using SQL for finding the similarity because relational database queries only give exact match answers. However, for Case-based Reasoning, the answer does not have to be a 100% match of a new case and an old case. Therefore, the use of SQL to determine the similarity of a new case and the stored old cases needs to define conditions or formats that support the use of SQL for finding the similarity of the new case and the old cases that cover all conditions.

We propose the use of SQL to find the similarity of a new case and the old cases stored in the database by presenting the cases with 100% similarity (Exact Match), with some similarity, and without similarity for the development of TQF Advisory System of the curriculum development.

2. Problem Statement of the research

At present, education is one of the businesses that are highly competitive because of modern technology that allows people the ability to choose various forms of education including in the education system at various institutions or to learn by themselves with various online media that provide vast and wide-open knowledge. The important thing of education, in addition to a teacher who has knowledge and the readiness and willingness to study of students, is the curriculum which can provide a framework or guideline for providing knowledge to students as well. Therefore, in the development or improvement of the curriculum, the developer can have a format for providing guidance which framework of concept should be developed in order to have graduates with knowledge exactly as specified.

The TQF Advisory System is an information system developed by using the Case-based reasoning technique in the development of the system in this work. The data of approved curricula are stored in a database and will be used as cases for various curricula. Curriculum developers can use them as a guideline for developing the curriculum by specifying the suitable direction of framework of each curriculum. For example, if a university wants to develop its computer science curriculum within the TQF standard, the system developer can refer to cases that are approved computer science curriculum from the database for curriculum development.

Using Case-based reasoning to develop a system can provide feedback to the curriculum developer through a case of the similar curriculum from other institutions in order to evaluate the developing curriculum. To determine the similarity of a new case and all old cases, the system of this work uses SQL programs to find the similarity.

3. Case-based reasoning in the research

Algorithm which are deployed in case-based reasoning (CBR) include nearest neighbor, induction, fuzzy logic and SQL retrieval.[3]

Nearest neighbor techniques are probably the most broadly utilized technology as a part of CBR since it is available in the majority of CBR tools. Nearest neighbor algorithms all work in a comparative manner. The similarity of the problem case to a case in a case-library for every case characteristic is resolved. This measure might be multiplied by a weighting component. At that point the total of the similarity of all attributes is determined to give a measure of the similarity of that case in the library to the objective case [3].

Induction techniques are generally utilized as a part of CBR since a lot of the more capable industrially accessible CBR tools give this facility. Induction algorithms, for example, ID3, fabricate decision trees from case histories. The induction algorithms recognize patterns amongst cases and separate the cases into bunches. Every bunch contains cases that are comparative. A prerequisite of induction is that one target case characteristic is defined. Basically, the induction algorithms are being utilized as classifiers to group comparable cases together. It is accepted that cases with comparative problem descriptions will allude to comparative problems and henceforth comparative solutions [3].

Fuzzy logics are a method for formalizing the typical handling of fuzzy linguistic terms, for example, excellent, good, fair, and poor, which are connected with differences in an attribute depicting a characteristic. Any number of linguistic terms can be utilized. Fuzzy logics inherently represent notions of similarity, because good is nearer (more similar) to excellent than it is to poor. For CBR, a fuzzy preference function can be utilized to compute the similarity of a single attribute of a case with the relating attribute of the target [3].

At its most straightforward form, CBR could be executed utilizing database technology. Databases are effective method for storing and recovering substantial volumes of information. In the event that problem descriptions could make well-formed questions it is clear to retrieve cases with identical descriptions. The issue with utilizing database technology for CBR is that databases retrieval using exact matches to the queries. This is ordinarily enlarged by utilizing wild cards, for example, “WESTp” matching on “WESTMINSTER” and “WESTON” or by specifying ranges, for example, “1965”. Using wild cards, Boolean terms and other operators within queries may cause a query more general, and subsequently more inclined to retrieve a suitable case, however it is not a measure of similarity. In any case, by increasing a database with clear knowledge of the relationship between concepts in a problem domain, it is conceivable to utilize SQL queries and measure similarity [3-5].

4. Similarity Retrieval

To determine the similarity of a new case at the parameter level, the procedure is as follows:

1. Specify the intended similarity range:

Table 1. Exact Matching and Level of Similarity Value

Exact Matching	Level of Similarity Value			Not Matching
	High	Medium	Low	
100%	100<Similarity value >=80	80<Similarity value>=60	60<Similarity value>0	Not found

Table 1 shows the specification of similarity between a new case and old cases from the answer from using SQL queries to determine the similarity including 1) if the answer for similarity between the new case and old case is 100% then the similarity value between the new case and old case is “Exact Matching”, 2) if the answer for similarity between the new case and old case is from 80% to under 100% then the similarity value between the new case and old case is “High”, 3) if the answer for similarity between the new case and old case is from 60% to under 80% then the similarity value between the new case and old case is “Medium”, 4) if the answer for similarity between the new case and old case is from over 0% to under 60% then the similarity value between the new case and old case is “Low”, and if there is no answer then the similarity value is “Not Found”.

2. Find the similarity value between the new case and old case using the following procedure:

   a. Specify a new case to determine the similarity value

   b. Specify a parameter for the new case in a) that will be used as a condition to determine the similarity value

   c. Find the similarity value of the new case by using the parameter from b) for comparing between the new case and old cases whether the new case is similar to which old case and at what similarity level as defined in Table 1.

   d. Store the obtained similarity value from 3) in a similarity table in the database.

The similarity of the new case and old case of the overall parameter in each side is determined as follows:

$$Similarity\ value = Max \begin{bmatrix} Max \begin{bmatrix} [Sum\ case\ 1(Parameter\ 1)] \\ [Sum\ case\ 1(Parameter\ 2)] \\ \vdots \\ [Sum\ case\ 1(Parameter\ n)] \end{bmatrix} \\ Max \begin{bmatrix} [Sum\ case\ 2(Parameter\ 1)] \\ [Sum\ case\ 2(Parameter\ 2)] \\ \vdots \\ [Sum\ case\ 2(Parameter\ n)] \end{bmatrix} \\ \vdots \\ Max \begin{bmatrix} [Sum\ case\ n(Parameter\ 1)] \\ [Sum\ case\ n(Parameter\ 2)] \\ \vdots \\ [Sum\ case\ n(Parameter\ n)] \end{bmatrix} \end{bmatrix}$$

The similarity value is determined from the similarity table obtained from the determination of similarity at the parameter level. To determine the similarity value at the overall level, the sum of each parameter in each case will be determined and then the highest sum of each parameter in each case will be determined. Finally, the highest sum of each parameter in each case will be used to determine the highest value in order to determine the similarity value.

5. The TQF Advisory System

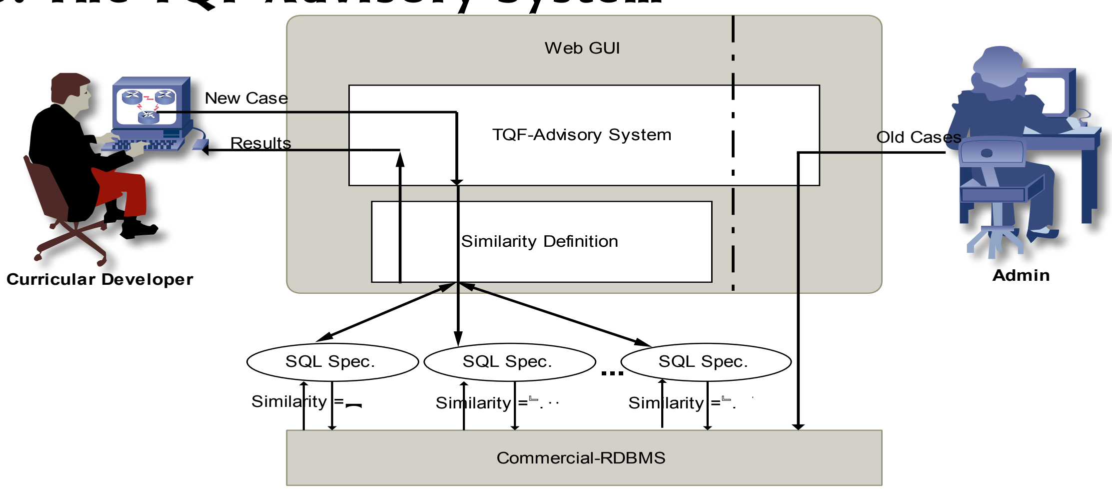

Figure 1. TQF Advisory System Architecture

The TQF Advisory System is a system that was developed as a tool for curriculum developer in higher education of Thailand to be able to see the overall picture that the curriculum prepared will eventually causes the students to possess knowledge in which direction under the 5 aspects of regulatory framework in each curriculum including 1) Ethics and Morality, 2) Knowledge Development, 3) Intellectual Development, 4) Interpersonal Relationship, and 5) Numerical Analysis, Communication, and Information

Technology Skills or the developed curriculum has the characteristic similar to the same curriculum from which institution so that it can be used as a reference or a guideline.

Figure1 shows TQF Advisory System architecture that illustrates the operation of the system, which includes two types of user. The first type is a user who is a curriculum developer, who is a user that use the work system to inspect or test the curriculum development that develops the characteristic of each curriculum or at the curriculum level it is similar to the same curriculum of which higher education institute in Thailand. To use it, user specifies the course information and the 5 aspects of responsibility information of the course whether it is a Primary Responsibility, Secondary Responsibility, or Not Relevant. The information that the user inputs will be compared with the information of the same course from other institutions that is stored in the database whether the course information in all five aspects of responsibility is the same as or similar to those of which institution. For the use in this part, user can update the curriculum mapping of the course according to user requirements to emphasize on any of the five characteristics, if found that the overall characteristic of the course is not consistent with the idea before storing that course into the database.

For the administrator (Admin), the admin will be responsible for storing the curriculum of each institute that has been developed and approved by the Office of the Higher Education Commission. Admin will store the details of curriculum mapping of each course of various curricula in the database.

Table 2. Example of Introduction to Database Curriculum Mapping **[6]**

Subject	Ethic and Morality							Knowledge Development								Intellectual Development				Interpersonal Relationship						Numerical Analysis			
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	8	1	2	3	4	1	2	3	4	5	6	1	2	3	4
8311202 Introduction to database	●	●	○	○			•	●		○	○		●		○	●	○		○	●		○		○		●	○		●

Table 2 shows the curriculum mapping of the course Introduction to Database that shows the 5 aspects of responsibility of the course that the curriculum developer specifies for students to be responsible. The solid circle represents the Primary Responsibility, the circle represents the Secondary Responsibility, and an empty slot means Not Relevant. TQF information are stored in the database whose schemas are shown in Figure 2.

University_Curriculum

U_ID	U_NAME

Subject

SubjectID	U_ID	SubjectName	Subject_Description	Unit

SCRV

Subject_ID	Characteristic	Responsibility	NValue

Case_Similarity

Subject_ID	Characteristic	USimilarity	NSimilarity

Figure 2. TQF Advisory System Relational Schema

The determination of the Similarity at the course level is a similarity comparison of a new course with courses stored in the database. The screenshot shown is shown in Figure 3 for the determination of the similarity. The procedure is as follows:

a) User specifies the course title in order for the system to find whether this course matches any course stored in the database. The result will show a description of the course along with the title for the user to double-check that the course is what the user needs. For example, if the user specify the name "Introduction to Database" the answer from the search system will show the course title with the words "Introduction to Database" and also show a description of all courses that are the answer in order for the user to verify that the specified title matches the course stored in the database.

b) The user specifies the parameters to be used as a condition for determining the similarity. For this research, the parameters used as condition in determining the similarity are the responsibility value of the course according to the characteristic in each aspect including Primary Responsibility (PR), Secondary Responsibility (SC), and Not Responsibility (NR).

c) The system will search the similarity of the course by using the course information from step a) and the responsibility parameters of the course from step b) to determine the similarity level of the above information with the courses stored in the database. The similarity level is classified into 4 levels including "Exact Matching", "High", "Medium", "Low", and "Not Found".

d) The results from step c) are stored in the Case_Similarity Table in the database which contains the subject ID (SubjectID), characteristic of the course (Characteristic), the names of the institutions with similar similarity value (USimilarity), and the similarity value of the course (nSimilarity) that the user specifies in a) the similarity value specified in b). A partial of a Similarity Table is shown in Table 3. The information obtained in this step will be later used as data for determining the similarity level of the curriculum.

Table 3. Similarity Table (partial)

Case_Similarity

SubjectID	Characteristic	USimilarity	NSimilarity
8311101	Ethics and Morality	U1	100
8311101	Intellectual Development	U1	86.66
8311101	Interpersonal Relationship	U3	86.66
8311101	Knowledge and Development	U2	73.33
8311101	Numerical Analysis	U1	73.33
8311107	Ethics and Morality	U1	80

Case_Similarity			
SubjectID	**Characteristic**	**USimilarity**	**NSimilarity**
8311107	Intellectual Development	U1	80
8311107	Interpersonal Relationship	U4	80
8311107	Knowledge and Development	U3	80
8311107	Numerical Analysis	U2	60

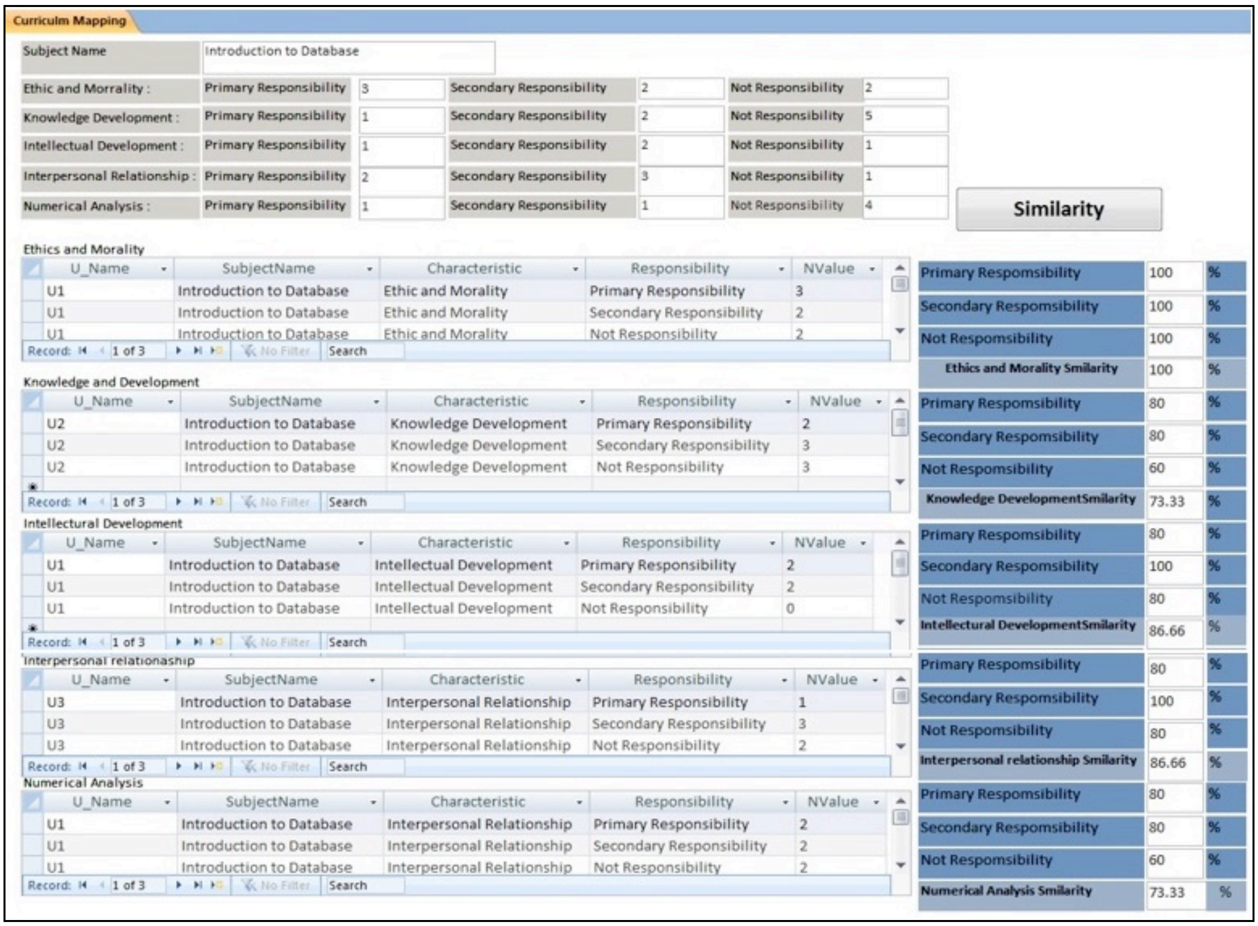

Figure 3. TQF-Advisory System Input Screen for Curricular Developer

Figure 3 shows the input screen of TQF Advisory System for the curriculum developer showing an example of finding the similarity of the course Introduction to Database which is a new case with some details explained as follows: Ethics and Morality of new case has similarity = 100% or "Exact Matching" with the course Introduction to Database of "U1" (old case), meaning the all responsibility values including Primary Responsibility, Secondary Responsibility, and Not Responsibility of both new case and old case are the same. For Knowledge Development of new case, the similarity = 73.33%

or the similarity level of "Medium" with the course Introduction to Database with "U2" when the Primary Responsibility of new case = 1 when the Primary Responsibility of old case = 2, Secondary Responsibility of new case = 2 when Secondary Responsibility of old case = 3, and the Not Responsibility of new case = 5 when Not Relevant of old case = 3.

The responsibility values for all aspects of each course that is a new case and the obtained similarity will be stored into the database. For this system, they are stored into the Case_Similarity table and when the system developer is satisfied with the result, the information of this new case will become an old case for use in the future.

The determination of the similarity at the curriculum level is a comparison of the new curriculum that user wants to find the similarity against curriculums stored in the database. The determination of the similarity at the curriculum level will use information on the similarity at the course level that is stored in the Similarity Table instead of taking stored data of all courses for the consideration. This is due to the fact that the information in the Similarity Table are cases which are the result from filtering out cases that are not relevant. So, accessing data for determining the similarity at this level is faster [7].
Figure 4 shows the SQL query which is used to determine the similarity at the curriculum level.

```
SELECT  Max(SumOfnSimilarity) AS MaxOfSumOfnSimilarity
FROM  (SELECT Case_Similarity.USimilarity,
Sum(Case_Similarity.nSimilarity) AS SumOfnSimilarity
FROM Case_Similarity
GROUP BY Case_Similarity.USimilarity);
```

Figure 4. SQL to find Curriculum Similarity

USimilarity	MaxOfSumOfnSimilarity
U1	419

Figure 5. Results from SQL in Figure 4.

Figure 5 shows the results for the determination of the similarity of the new curriculum (new case) against the existing curriculum (old case) of institution "U1" when considering the similarity of the characteristic of all courses at all aspects of characteristic and the curriculum of "U1" has the highest similarity. Therefore, it can be concluded that the new curriculum that is being developed (new case) is similar to the same curriculum of institution "U1".

SQL statements for finding the similarity of the curriculum in the overall characteristic of each aspect of the curriculum are shown in Figure 6.

Curriculum Characteristic	SQL for finding Curriculum Similarity for a curriculum characteristic
Ethics and Morality	SELECT USimilarity, Max(SumOfnSimilarity) As Max**OfnSimilarity**

	FROM (SELECT Case_Similarity.USimilarity, Sum(Case_Similarity.nSimilarity) AS SumOfnSimilarity, Case_Similarity.Characteristic FROM Case_Similarity GROUP BY Case_Similarity.USimilarity, Case_Similarity.Characteristic HAVING (((Case_Similarity.Characteristic)="Ethics And Morality")) ORDER BY Sum(Case_Similarity.nSimilarity) DESC);
Knowledge Development	SELECT USimilarity, Max(SumOfnSimilarity) As Max**OfnSimilarity** FROM (SELECT Case_Similarity.USimilarity, Sum(Case_Similarity.nSimilarity) AS SumOfnSimilarity, Case_Similarity.Characteristic FROM Case_Similarity GROUP BY Case_Similarity.USimilarity, Case_Similarity.Characteristic HAVING (((Case_Similarity.Characteristic)="Knowledge Development")) ORDER BY Sum(Case_Similarity.nSimilarity) DESC);
Intellectual Development	SELECT USimilarity, Max(SumOfnSimilarity) As Max**OfnSimilarity** FROM (SELECT Case_Similarity.USimilarity, Sum(Case_Similarity.nSimilarity) AS SumOfnSimilarity, Case_Similarity.Characteristic FROM Case_Similarity GROUP BY Case_Similarity.USimilarity, Case_Similarity.Characteristic HAVING (((Case_Similarity.Characteristic)="Intellectual Development")) ORDER BY Sum(Case_Similarity.nSimilarity) DESC);
Interpersonal Relationship	SELECT USimilarity, Max(SumOfnSimilarity) As Max**OfnSimilarity** FROM (SELECT Case_Similarity.USimilarity, Sum(Case_Similarity.nSimilarity) AS SumOfnSimilarity, Case_Similarity.Characteristic FROM Case_Similarity GROUP BY Case_Similarity.USimilarity, Case_Similarity.Characteristic HAVING (((Case_Similarity.Characteristic)="Interpersonal Relationship")) ORDER BY Sum(Case_Similarity.nSimilarity) DESC);
Numerical Analysis	SELECT USimilarity, Max(SumOfnSimilarity) As Max**OfnSimilarity** FROM (SELECT Case_Similarity.USimilarity, Sum(Case_Similarity.nSimilarity) AS SumOfnSimilarity, Case_Similarity.Characteristic FROM Case_Similarity GROUP BY Case_Similarity.USimilarity, Case_Similarity.Characteristic HAVING (((Case_Similarity.Characteristic)="Numerical Analysis")) ORDER BY Sum(Case_Similarity.nSimilarity) DESC);

Figure 6. SQL statements for finding the similarity of the curriculum in the overall characteristic of each aspect of the curriculum

6. Conclusion

Case-Based Reasoning is a methodology widely used in many fields such as education, tourism, medicine, or other industries because it is a method that uses data from the past to support the decision making for the present situation like TQF Advisory System which is an information system developed using Case-Based Reasoning and SQL to find the similarity level between new case and old cases both at the course and curriculum levels. Users can use the data from this work as a guideline for the development of a curriculum in order to choose the direction of the development.

7. References

Aamodt, A., and Plaza, E.: 'Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches', AI Communications, 1994, 7, (1), pp. 39-59

Kolodner, J.L.: 'An Introduction to Case-Based Reasoning', Artificial Intelligence Review, 1992, 6, pp. 3-34

Watson, I.: 'Case-Based Reasoning is a methodology not a technology', Knowledge-Based System, 1999, 12, pp. 303-308

Portinale, L., and Verrua, A.: 'Exploiting Fuzzy-SQL in Case-Based Reasoning'. Proc. The Fourteenth International Florida Artificial Intelligence Research Society Conference, Key West, Florida, USA2001 pp. Pages

Schumacher, J., and Bergmann, R.: 'An Efficient Approach to Similarity-Based Retrieval on Top of Relational Databaases', Advances in Case-Based Reasoning, 2003, 1898 of the series Lecture Notes in Computer Science, pp. 273-285

OHEC: 'National Qualifications Framework for Higher Education in Thailand', in Editor (Ed.)^(Eds.): 'Book National Qualifications Framework for Higher Education in Thailand' (2006, edn.), pp. 1-27

Burkhard, H.-D.: 'Case Completion and Similarity in Case-Based Reasoning', ComSIS, 2004, 1, (2), pp. 27-55

Contact email: proud.it.pkru@hotmail.com, suphamit.ch@kmitl.ac.th