DOCUMENT RESUME

ED 428 707 IR 019 368

AUTHOR	Okamoto, Toshio; Nakagawa, Masaki
TITLE	A Case Base Reasoning System on the Internet for Reference of Information Technology Education for Teachers.
PUB DATE	1998-06-00
NOTE	7p.; In: ED-MEDIA/ED-TELECOM 98 World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications. Proceedings (10th, Freiburg, Germany, June 20-25, 1998); see IR 019 307. Some figures may not reproduce clearly.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Access to Information; *Case Method (Teaching Technique); *Class Activities; Computer Software Development; Computer System Design; Computer Uses in Education; Databases; Educational Technology; Foreign Countries; High Schools; Indexing; Information Retrieval; *Information Science Education; Information Technology; Lesson Plans; *Online Systems; Secondary School Teachers; Shared Resources and Services; *Teaching Methods; World Wide Web
IDENTIFIERS	Japan

ABSTRACT

This paper describes the development of a case-based system for information technology (IT) education, to be used in high schools across Japan, so that teachers can share a common knowledge about IT education. The first section covers the study purpose and describes the case-based system, including registration of cases, searching cases, and adjusting search results. The use of case-based reasoning (CBR) theory in the construction of educational systems is described in the second section. The third section outlines the following components of the system architecture: use of the World Wide Web; the case base, including profile, contents, and address descriptors; the interface; case registration; and case search, including approximate cases and the similarity function. The final section explains the process of CBR learning in more detail, including case modification, case diagnosis, and case repair. Three tables present possible values for profile, contents, and address descriptors. (DLS)

A Case Base Reasoning System on the Internet for Reference of Information Technology Education for Teachers

Toshio Okamoto & Masaki Nakagawa
Graduated School of Information Systems, University of Electro-Communication, JAPAN.
e-mail: {okamoto,chusen} @ai.is.uec.ac.jp

Abstract: While our society is becoming increasingly information-oriented, Information Technology Education (IT-Education) has just started from elementary schools to highschools, in nearly all Education sites. We are building a database which covers actual cases of such education, together with a management system based on CBR (cased based reasoning), accessible to the public through the Internet. Teachers can input their experience of IT-Education into this system using the CBR method, or just check among past cases, similar to their present application. By using this system, teachers from any Education environment, urban or local, can share a common knowledge of IT-Education practice.

Foreword

The modern society is more and more information oriented, thanks to the advances of scientific technology we witness in everyday life. We accept this as a fact. This phenomenon has brought about a new kind of teaching which not enough history to refer to and apply in actual class teaching. The existing cases are in reality not known to the public, and there seems to be no market to exchange the ideas and information about such experiments. They are hidden as teachers' private experiences. It is therefore required to create a place where private knowledge may be shared by a wider group of teachers through IT-Education and lead to a global refining of teaching methods, which would lead to quality improvement and to teachers' self improvement [Okamoto 1996, Nishinosono 1996]. Our aim with this research is to construct a Case-Based System for Information Technology being taught at high schools nationwide and put the developed system on the Internet. A great number of teachers could access this system, however far they are located from each other geographically, and use this system anytime in spite of hindrances such as time and location.

Study Purpose and Case-Based System

The users of our system will be teachers who are actually contributing to IT-Education, and also would-be participants. The system enables us to build a database for actual practices of IT-Education, and also accepts searching and registration of cases from anyone. For those who are actively involved in the Education process, reference to past similar cases will help to improve their teaching methods, when they find their teaching process not going as well as they've expected. For the would-be participants the system will offer past good and bad examples of the area, as guiding points for their future research. Our system sear the past history of the cases similar to a particular or possible class lesson, given some profile descriptors as IT-Education search criteria.

Registration of Cases

Registration of cases can be done by any teachers who have experience in teaching IT-Education. Data are input by several formats. Class, teaching method are input by choosing among a set of optional, ready-made items. At the same time, teachers can fill in the free area with their own work-out programs and students' reactions without following any specific format. Also, they can put levels of relative value between input cases and actual teaching contents by adjusting some tuning options. Thus the system learns by acquiring new knowledge with every registration of actual teaching contents and methods. Registered cases are to be included in the database and can be a target for future search.

Search of Cases

Case searching is designed for teachers who need to check past cases of IT-Education. They will input profile descriptors which define search patterns. The system actually sear past cases using these search patterns and a similarity measurement, and finally returns the case(s) which match the tea need best. If the result corresponds, in the system's conception, to the actual need of the teacher, the system will display it. The result contains information about the actual class procedure, about the topics and about an overall judgement of the teaching process. If the teacher wishes, he can ask for further detailed information about each item of these cases.

Adjustment of Cases

If the output is not what the user wants, the system tries to adjust the result by using the rule base of domain knowledge that is pre-installed, to fit the user's need approximately. When this adjusted result is considered to be appropriate, according to the similarity function computation, it is called a 'hit', and the adjusted result is added to the database as a new case.

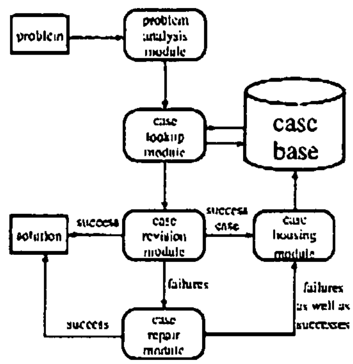

Figure 2 : A typical example of CR

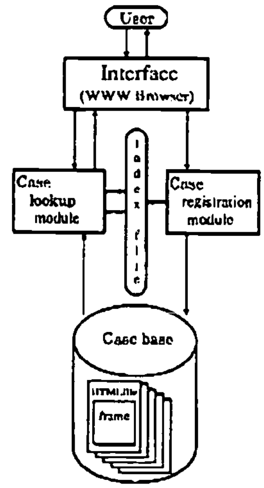

Figure 1 : The system architecture

Figure 2 : A typical example of CR

Figure 1 : The system architecture

CBR

To construct a system as above, CBR theory of knowledge processing and technology is used [Riesbeck 1989, Hammond 1990, Kobayashi 1992]. We give here the details of CBR and explain its efficiency for our system. Expert systems tools such as Rule Based Reasoning and Model Based Reasoning have been widely used to solve problems within certain intelligent systems. The former is efficient as it is based upon the expertise of domain specialists, but is vulnerable to knowledge acquiring. Similarly, Model Based Reasoning, which uses principles and laws of the structure of the target system, shows high reliability, but low efficiency. CBR consists of a set of problem solving architectures leading to a solution, wile using past cases of success and failure, similar to a given problem [Hammond 1989]. A typical example of CBR architecture is shown in fig.1. CBR eliminates the above problems and adds efficiency.

An exhaustive search through the case base takes a lot of time and a vast amount of calculation. CBR can reduce the tracking space by applying filtering criteria directly to an index file, which saves search time and volume. Therefore, our system can be implemented on middle range computers. When the picked-up result match the problem, the result is the desired solution. The result can be, however, rather far-fetched. Thus the need of adjustment using appropriate domain knowledge with the case repair module, which finally leads to the desired solution. This adjusted result is applied to the particular problem, and if it solves it, it is added to the database as a successful case [Kolodner 1993].

System Architecture

The Case Based System is being developed with the use of the WWW so that teachers and common users spread far from each other geographically may have access to the same system. The system doesn't require specific browsers, but can function on any desired one, and offers search and registration service at anytime and in anyplace. Users' information is processed and administrated systematically with the use of the CGI (Common Gateway System).

The WWW interface offers an interactive process between users and the system. cases are kept as HTML files. For efficiency's sake, an index file keeps profile descriptors of the cases. This file is used when the actual searching takes place. The system architecture is shown in Fig.2.

Case Base

The Case Base comprises two kinds of files. One is a set of case base files which describe actual class cases in the form of a frame, and the other one is an index file, which keeps specific descriptors of actual class cases. When a case is registered as a HTML file, registration/installation is done in the form of a frame. Three descriptors create each frame: the profile descriptor, the contents descriptor and the address descriptor.

Profile Descriptor

The profile descriptor contains the characteristic elements of actual class cases. The practice of IT-Education varies much, depending on the characteristics of each class and on the thematic genre of the specific informational area. The teaching procedure is selected by evaluating both class and specific thematic genre. Computers can be used in a class in a variety of ways. It is presumed that a class, where students actually write programs using computers, gives a different result from when they learn with the aid of ready-made applications such as CAI programs. If computers are used at an inappropriate time during class, we cannot expect good results. We also have to take into consideration that the number of computers available varies from school to school, so that the number of computers available to one student depends on the school the student belongs to. Therefore, the result from using computers is not simple. Also, the teaching/learning procedure is influenced by the different class aims, so that even if the same thematic genre of informational area would be studied, but in different classes with different subjects, the result cannot be the same. With all these circumstances in mind, we defined the profile descriptor slots as follows. Table 1 shows the profile descriptor slots and their possible values (range facets).

Table 1 : Profile Descriptors; some values

Slot names:		Range facets:
Pin-pointed	I) Teaching Subject	Simulation, Internet, Spread Sheet, Graphic Analysis, Programming, Chart Making, Problem Solving and its Algorithm, Measurement and Control of Information, Basic information in Technology, Information and Society
	II) Names of classes	English, Math., Language, Science, Social Study, Art, PE.
1) Lowest relation degree allowed between Teaching Subject and database case		Comparison and relativity value between teaching contents and related output (0-10)
Class working rate		2) Rate of Lectures, 3)Practice, 4)Experiment/lab. Work in terms of time (0-10 respectively)
Class formation rate		5) Individual learning, 6) Simultaneous learning, 7) Group learning. Their rate in terms of time(0-10 respectively)
8) Frequency of computer use		Use of computers during a class, in terms of time percentage (0-100%)
9) Use of computers		For word processing, for CAI, for communication as a notice board for students, for the Internet, for CAD, or no CAD at all
10) Number of students registered for a class		Actual number of students (no upper limit)
11) Number of computers		Number of available computers (no upper limit)

Contents Descriptor

The contents descriptor shows the details of the teaching contents and the teaching result of each case. The IT-Education knowledge of teachers who have actually participated in the teaching is to be assessed with a value in each slot. Teachers are expected to offer a teaching subject with an appropriate Education purpose and to add also the teaching contents. They have to study the essentials of a certain teaching content and use a variety of teaching materials and machinery devices, in order to create a lesson easily understood by the students. At the end of each class session, teachers grade students from the various points of view of the IT-Education. Moreover, teachers assess their own IT-Education ability themselves, as being either a good or a bad teaching job, and they compile the important points of the education process and add them to the IT-Education curriculum. With these in mind, we designed the slot column and value column of the contents descriptor. Table 2 shows examples of slot and value fields of this descriptor. For the value field, whole text files can be used to fill in the column, so that there is enough place for detailed information.

Address descriptor

This descriptor shows the detailed environmental information of each class teaching: school name, where the teaching has been carried out, teacher's name(s), class name(s) (non-specific, science, English, economy, engineering, agriculture, polytechnic, etc.), class tag and students' grade. At the same time, when the school's network topology and the detailed teaching information at that school can be reached through the Internet, the

Table 2 : Contents Descriptors; values

Slot names:	Range facets:
Aim of class	To know traditional local industory and ...
Characteristics of the class study	Referring to traditional local industory through a homepage .
The good points of the class	Using the Internet ...
Teaching suject	Traditional local products
Teaching device	Video tapes
Accumulated knowledge presentation	Putting the student papers on a homepage ...
Points to be amended; Proposals to curriculum	The number of computers per student are limited .....
General report on the result	More interest for the internet was achieved ...
Grade	Good job

Table 3 : Address Descriptors; some values

Slot names:	Range facets:
School	University of Electro-Communications, Tokyo
Teacher in charge	Toshio Okamoto
Genre	Engineering
Class Tag	Compulsory
Grade	Second year
Network topology	Has access to Internet
URL	http://www.ai.is.uec.ac.jp/

URL of the school is given. Table 3 shows an example of slots and values of the address descriptor.

Interface

The system http://www.ai.is.uec.ac.jp/~chusen/CBR/SYS/ offers two interfaces, the one when users register, and the other one when users search for similar past cases. To make a registration, users have to fill in the above three descriptors, i.e., profile, contents and address descriptor. Input data will be transferred to the registration module of the system server through the Internet. Fig.3 shows the registration menu.

Searching for past cases, the users input characteristics of the data they need, as value columns of the previously shown profile descriptor. Users can input searching weights values by manipulating keywords (Fig.4). As users have three possible outcomes for their search: past successes, failures, and a mixture between the two, they can specify which of these results are to be shown to them Users can be interested in past failures as well as in past successes, depending on their interest in reiterating success cases or preventing mistakes that have been already made. .

Case Registration

The case registration module records values from the above stated three descriptors into one HTML file. At the same time, the system draws a numerical value from the key descriptors, and installs a numerical value in the index file. The index file is there to make case search more efficient. In the registration process, the inputs with no valid value are automatically rejected.

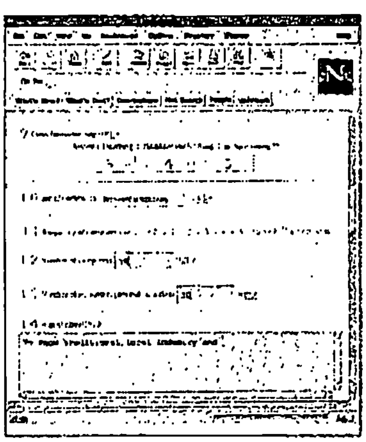

Figure 3 : Registration Page

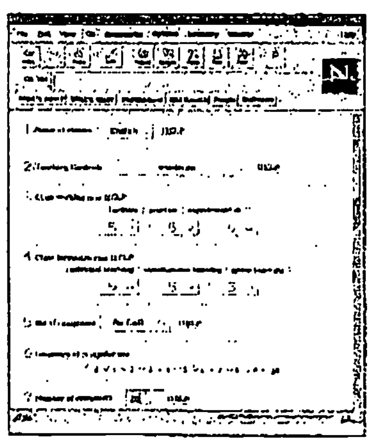

Figure 4 : Searching Page

Case Search

The case search module draws search patterns from the information clients have given. These patterns include keywords from the key descriptors' value column and also search formats. Search formats, class genre and class contents determine the system values of the appropriate slot. The system has thus the possibility to choose suitable cases matching all these search conditions from the index file. The system then inputs these similar cases into a similarity function and draws as satisfactory cases as possible from among them. Thus, the output result cases drawn from the vast database are considered to satisfy the search condition best.

Approximate cases

The system starts searching for patterns similar to those input by the user. To apply the similarity function, which can be also seen as a search function, to all the cases in the database, means scanning a vast search space, which can be very inefficient and time-con Therefore, to minimise the search area, the system pinpoints approximate cases with the aid of input values from two kinds of slots here, for instance, slot 'teaching subject' and slot 'name of class' of the profile descriptor. Class names and teaching contents are indispensable search patterns to pinpoint cases. The system will then apply the similarity function to the pinpointed cases, and then will check the final output value of the similarity function. The case with the minimal distance between users input and registered data are considered to be the best result with the highest similarity degree to the user's needs. This pinpoint procedure enables the system to seek through as little case space as possible for the target and leads to an efficient case search.

Similarity Function

The system carries out pinpointing with a matching process referring to the two highly significant slots mentioned before and applies the similarity function to each of the pinpointed cases. This is how the similarity value is computed. The system checks the similarity value to measure the distance between each case and the search pattern input by the user. The shortest distance between them will be the best approximate case. We define the similarity function as:

$$D(U,T) = 1/9\left[\sum_{i=1}^{7}(Su_i - St_i)^2 + (1 - Su_{key})^2 + d(Su_9, St_9) \times Comp\right]$$

$$Comp = |Su_8 - St_8| \times (1 - Com_{rate})$$

$$Com_{rate} = \begin{cases} \frac{Su_{11}/Su_{10}}{St_{11}/St_{10}} & (Su_{11}/Su_{10} < St_{11}/St_{10}) \\ \frac{St_{11}/St_{10}}{Su_{11}/Su_{10}} & (Su_{11}/Su_{10} > St_{11}/St_{10}) \end{cases}$$

$$Su_{key} = \frac{key \times n}{St_{result}}; \quad d(u_i, v_i) = \begin{cases} 1 & (u_i = v_i) \\ 0 & (u_i \neq v_i) \end{cases}$$

Here, U is the search pattern users input. T is a case already in the database. $D(U,T)$ is the distance between U and T. Su_i is the numerical value of the search pattern, St_i the numerical value of the registered case, with i the order of the numerical search pattern in the profile descriptor (Table 1).

Su_{key} is the percentage rate of the input keyword in the value slot corresponding to the slot 'General report on the result' in the contents descriptor (Table 2), with *key* the number of letters of the desired pattern, n the number of occurencies of the desired pattern in the text, and St_{result} the total number of letters in the database text. As for Su_{key}, an example of computation can be found in Fig.5.

The slots (1)-(11) in Table 1 are used for the computation of the similarity function used in the searching

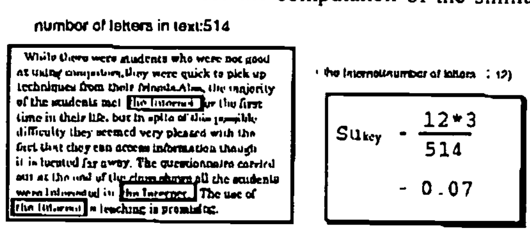

Figure 5 : Example of Su_{key}

process. Each value of (1)-(7) is input by the user as values from 0 to 10. These inputs are converted into 0-1 values, and handed over to the similarity function. The evaluation of each slot (1)-(7) means the computation of the Euclidean distance between the searching conditions input by the user and the recorded values. In what concerns slots (8)-(11), they are evaluated from the point of view of *computer usage* and integrated into a single value, the *Comp* value in the similarity function definition equation. $d(u_i,v_i)$ computes the difference between the stored pattern and the user's input for the usage purpose of the computer (contents of slot (9), profile descriptor, Table 1). Here, $d(u_i,v_i)$ is computed as follows: it is 1, if u_i and v_i values coincide, or 0 if not. (10),(11) are used for the computer per student rate, and generate the value Com_{rate}. After computing this presented function, the system returns to the user the result which has the lowest $D(U,T)$ value.

Learning

In this chapter we explain in more detail the process of CBR learning. The main system's training consists in the usage of the case modification mechanism, the case diagnosis mechanism and the case repair mechanism.

Case Modification

In the case searching process, the most similar case, according to the user's search conditions, is gathered. The profile descriptor of the searched case and the user input search conditions are rarely in perfect concordance. Therefore, cases which don't coincide can be tuned by the modification mechanism, according to the user's conditions.
The user is tuning all slots for the case he is looking for. During tuning, the system presents pertinent choices for the slot, so the tuning can be done easily.

Case Diagnosis

The case diagnosis mechanism is testing the validity of the case which was tuned. The diagnosis mechanism is using the domain knowledge base of relations between the slots or the slots' value, a.s.o. If the result of this diagnosis is valid, the case is presented to the user. If it is not valid, the reason of invalidity is shown to the user, and a new tuning is repeated.

Case Repair

After the case diagnosis, the case diagnosed is attached features with the case repair mechanism, that are stored in the case data base. The repair mechanism adds some information and saves it as a case repair information frame. The stored information are 1) an information pointer to the modified case (the most similar case to the user's search conditions) and 2) the case modification frequency flag. This information is used for the case search. In the process of case selection, the system sear the case input by the user (we call this case the original case). If nothing matching is found, the system sear the modified case base, based on the original case and the case modification frequency flag. This flag expresses a class hierarchy number, that is created together with the modified case, and is equal to the number of modifications of the original case. Therefore, the functioning of the search mechanism reflects the modification experience.

Discussion

As the system requires from users to specify the desired relative value between input case and the actual teaching subjects stored in the database, a precise assessment. The fact that desired patterns that have to appear in the teaching contents allow a more flexible search than simple keyword matching systems.
The system has now been actually implemented and can accept case registration and provides the search function. Future improvement can be done to adjust the output result.
We are installing one case as one HTML file just to make the search procedure efficient, but it is presumed that the registration number will increase to a vast degree. It is impossible to control the whole database by only one domain frame. Therefore, we are planning to split the database into several different divisions, so that a distributed management system may be constructed. After all these adjustments, the system will be able to offer better results to the users.

Reference

[Okamoto 1996] Okamoto, T. *et al.* (1996), Feasibility. Study for a practice of IT-Education in Sencond High school and it's evaluation , (res. no.: 07308017), yr.7(Jap.).

[Nishinosono 1996] Nishinosono, H. (1996), IT-Education Curricilum Development for General Course in High School, Science Education, Vol.20,No.1, pp.42-50.

[Riesbeck 1989] Riesbeck, C.K. and Schank, R.C. (1989), Inside CBR , Lawrence Erboum Assoc..

[Hammond 1990] Hammond, K.J. (1990), Case-based Planning: a framework for planning from experience , Cogn.Sc., Vol.14, pp.385-443.

[Kobayashi 1992] Kobayashi S. (1992), Pres. of CBR , Journ.JSAI,Vol.7, No.4, pp.559-565.

[Hammond 1989] Hammond,K.J. (1989), Case-based Planning as a memorial task , Academic Press.

[Kolodner *et al.* 1993] Kolodner,J. and Riesbeck, C. (1993), Case-Based Reasoning , Morgan Kaufman Publisher.

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").