Making Static Online Teaching Materials Be Flexible to Learners by Reconstructing Its Hypermedia Structures Automatically

Athena Hsieh, Rita Kuo, Chang-Kai Hsu, Maiga Chang, *Member,IEEE*, and Jia-Sheng Heh

***Abstract*—Many teachers and researchers put their teaching materials on the Internet for students to read in recent years. This sort of teaching materials could be seen as static because students can only follow the learning sequence made by teachers in advance. The goal of this paper is trying to develop a tool, K-Navi toolbar, to parse and rebuild teaching materials' hypermedia structures according to the concept relations and extracted rules automatically. K-Navi toolbar first uses the Formal Concept Analysis (FCA) to parse the whole set of teaching materials and gets the embedded concept relations between each of two instructional documents; then uses the Association Rule Methodology (ARM) to extract the rules from the concept lattices; and, finally rebuilds the hypermedia structure of the instructional document read by the student automatically. K-Navi toolbar adds related concept hyperlinks (or says links to other knowledge pieces) into the specific position on the instructional hypermedia document automatically when a student asks the document resource. A student then will be able to dig related knowledge pieces via these relevant hyperlinks.**

***Index Terms*— e-Learning, formal concept analysis, association rule, keyword, knowledge navigation, hypermedia environment, adaptive web**

I. Introduction

Chen (2001) thoughts that Internet-based Education allows students to search and retrieve the teaching materials which they wanted to learn via information communication technologies (ICT)[4]. And Rosenberg (2001) proposed that teachers can upgrade, deposit and withdraw their teaching material on internet. Because of network, it is easy to spread and share the content of courses and information. Besides, teachers also can examine the effect of students' learning via network [2] [3] [5]. With the results, that helps them know students' learning more and teach more effectively.

This paper tries to present a reconstruction mechanism in order to rebuild the hypermedia structure of instructional materials by adding relevant hyperlinks to connect related teaching materials. With those hyperlinks students can learn more about specific keywords which they are interested in, even those hyperlinks do not exist on the original webpage. For reaching the goal, we should take data mining techniques into our considerations when we design the hypermedia structure reconstruction flow and related tool, the K-Navi toolbar. The tool can find concept associations; reconstruct the instructional materials; and, add the relevant hyperlinks to a static webpage. The tool would make students learn more effectively.

Section 2 describes related theories and techniques what we can apply into the hypermedia tool development and make static hypermedia resources become flexible to users. In Section 3, a series of examples give us a view of what the hypermedia structure reconstruction flow is and how we plan to realize such system. Section 4 describes the hypermedia structure reconstruction flow from a student visits the learning website to he/she sees the redecorated webpage with some relevant hyperlinks that had never been saw before. An experiment tool, K-Navi toolbar, is implemented for evaluating the association lattice and rule extraction approach in Section 5. Section 6 makes a simple conclusion and discusses possible future works.

II. Adaptive Hypermedia Environment

The goal of this paper is trying to develop a tool, K-Navi toolbar, to parse and rebuild teaching materials' hypermedia structures according to the concept relations and extracted rules automatically. K-Navi toolbar first uses the formal concept analysis (FCA)[6] to parse the whole set of teaching materials and gets the embedded concept relations between each of two instructional documents; then uses the association rule methodology (ARM) to extract the rules from the concept relations; and, finally rebuilds the hypermedia structure of the

Manuscript received April 07, 2006. This work was supported by the the National Science Council of the Republic of China under Contract No. NSC 94-2520-S-033-003.

Athena Hsieh is with the Department of Information and Computer Engineering, Chung-Yuan Christian University, Chung-Li, 32023 Taiwan (e-mail: s9227138@cycu.edu.tw).

Rita Kuo is with the Department of Electronic Engineering, Chung-Yuan Christian University, Chung-Li, 32023 Taiwan (e-mail: rita@mcsl.ice.cycu.edu.tw).

Chang-Kai Hsu is with the Department of Electronic Engineering, Chung-Yuan Christian University, Chung-Li, 32023 Taiwan (e-mail: ken@mcsl.ice.cycu.edu.tw).

Maiga Chang is with the Program Office of National Science and Technology Program for e-Learning, Chung-Li, 32001 Taiwan (e-mail: maiga@ms2.hinet.net).

Jia-Sheng Heh, is with the Department of Information and Computer Engineering, Chung-Yuan Christian University, Chung-Li, 32023 Taiwan (e-mail:jsheh@ice.cycu.edu.tw).

1-4244-0406-1/06/$20.00 ©2006 IEEE.

instructional document read by the student automatically.

A. Hypermedia Environment

Users might read hypermedia from one page to another page through hyperlink via browsers arbitrarily [11]. It leads distance learning on WWW to asynchronous mode. Teachers will not be able to know whether students read these documents orderly or not [1]. In this way, teachers do not know what each student has learned and might not able to give student suitable help. In addition, students will not know how to deal with the difficulty either.

As we know that there are more and more teachers and researchers put their instructional matters on the Internet for students to read. Although Internet has brought a lot of benefits, some teaching materials on the Internet could still be seen as static because students only can follow the learning sequence made by teachers in advance. Students can not learn more about what they are really interested in. For example, Ken wants to know more about the concept, 'Object', while he reads the 'Class' section on the Sun Java Tutorial website http://java.sun.com/docs/books/tutorial/java/concepts/class.html.

Ken will never get what he wants because the tutorial website designers had never considered that the student might want to know the 'Object' while he/she reads the teaching materials about 'Class' when they built the Sun Java Tutorial website.

Therefore, applying knowledge structure to modify hypermedia structure of instructional materials is necessary. Because knowledge structure could help students to retrieve related information and understand learning status.

Here comes an important issue: how to supervise a student's learning path in a non-monitored e-Learning environment? We need to give the student a right and good instruction for the next document to read (or says to retrieve). Thus, the student might study more easily and efficiently if we can design knowledge structure with several knowledge operations; generate knowledge navigation rules; and create reference links in a hypermedia document to other relevant documents automatically.

B. Formal Concept Analysis

Formal concept analysis is a mathematical theory based on ordered set and complete lattice. And it is also a method mainly used for the data analysis [7][9]. In the Java language domain, the most well-known concepts for instance are "Object", "Class", "Method", "Inheritance", and "Variable". For example, there are six document structures as Table I lists. At first sight, we might not find out the relation of the concepts listed in Table I. But we can create formal context from Table I based on FCA theory.

Table II shows a simple example formal context of Java course. First, we let the object set O is $\{s_1, s_2, s_3, s_4, s_5, s_6\}$ and the attribute set A is {O, C, M, I, V}. We know that the keyword "Object" appears in the s_1, s_3, s_4, and s_5. We can use the formal context to represent the corresponding knowledge (keywords with documents) as Table II lists.

TABLE I
FIVE KEYWORDS IN SIX DOCUMENT STRUCTURES.

s_i	Keywords in s_i
1	OCIV
2	CMV
3	OCIV
4	OCMV
5	OCMIV
6	CMI

TABLE II
SIX DOCUMENTS OF JAVA COURSE AS A FORMAL CONTEXT.

	s_1	s_2	s_3	s_4	s_5	s_6
O	x		x	x	x	
C	x	x	x	x	x	x
M		x		x	x	x
I	x		x		x	x
V	x	x	x	x	x	

After the formal context has been created, a special concept hierarchy likes Fig. 1 can be draw out from the lattice produced by FCA.

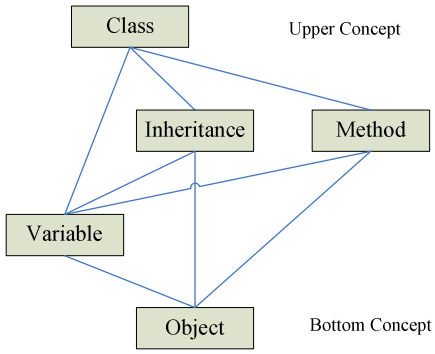


Fig. 1. The corresponding keyword hierarchy of ontological concepts.

Fig. 1 points out that the concept "Class" is an upper bound of three concepts: "Variable", "Inheritance", and "Method". In the other words, concept "Class" is more representative than all those five concepts. Furthermore, "Class" will be the basis concept of the course, "Object Oriented Programming in Java".

C. Rule Extraction

We can take any structured documents in the hypermedia environment as transactions in commerce. Moreover, those keywords in the documents could be also seen as the frequent purchasing items of customers, and the customers could be either documents or students in a hypermedia environment.

Association rule mining (ARM) is a well explored research area, we will only introduce some basic and classic approaches for association rule mining. The Apriori [8] algorithm is the most well-known association rule algorithm and is used in most commercial products. Apriori is just a straightforward approach that requires many passes over the database, generating many candidate itemsets and storing counts of each candidate while

most of them turn out to be non-frequent.

To show that how the rule extraction works, consider the transaction set example given in Table III. Based on these transactions and the minimum support threshold, minsup = 80%, the frequent itemsets were generated in Table IV.

TABLE III.
SAMPLE TRANSACTIONS INVOLVING ITEMS A, B, C, D AND E.

Transaction	Items
t_1	ABDE
t_2	ABECD
t_3	ABEC
t_4	BEBAC
t_5	DABEC

TABLE IV.
ALL FREQUENT ITEMSETS WITH MINIMUM SUPPORT = 80%.

Support	Itemsets
100% (5/5)	{A},{B}, {E}, {AB},{AE},{BE}, {ABE}
80% (4/5)	{C}, {AC},{BC},{CE}, {ABC},{ACE},{BCE}, {ABCE}

Take the frequent itemset {BE} for example, since both {BE} and {A} appear in every transactions, a rule, $\{BE\} \xrightarrow{1.0} \{A\}$, can be extracted. Similarly, because {C} and {AC} only appear in 80% transactions, which means, the other two rules, $\{BE\} \xrightarrow{0.8} \{C\}$ and $\{BE\} \xrightarrow{0.8} \{AC\}$, are sustained.

III. A COMPLETE EXAMPLE

A hypermedia instructional material is composed of a keyword set and several document structures. The keyword set can be seen as the itemset in Apriori algorithm. When we want to extract some ***confident association rules***, we need to set minimum support and confidence thresholds first. The confidence threshold helps us to reconstruct the original webpage to a really useful one. In this Section, a series of examples give us a view of what the hypermedia structure reconstruction flow is and how we plan to realize such system.

Nelson(1995) suggested an idea about associating hypertext in a multiple and flexible way for any hypermedia[10]. Network structure is one possible metaphor to represent the information of nodes and links in a hypermedia environment. No matter a structured document is hypertext or not, which can be transformed into the independent hierarchical (no-flat) structures, is so-called the ***document structure***, $S = \{s_i\}$. The elements of the document structure include chapters, sections, paragraphs, and sentences.

Example 1. Fig.2. shows an example of hypermedia environment. Here, the ***document structure*** s_1 (e.g., article) has a header associated with the keyword k_1 and a paragraph S_{11}. The structure s_{11} also has another header associated with the keyword k_2, one sentence (s_{111}), and one hyperlink to the document s_2. The sentence s_{111} contains two keywords, they are k_3 and k_4. The document s_2 contains another two keywords, k_5 and k_6. □

In the most of e-Learning systems, webpage designers often focus on the content, such as pictures, animation objects, video games, and online-test. However, they seldom pay an attention to consider the students' possible traversal paths. Different traversal paths may make students feel comfortable or confused, especially when the instruction materials are designed for a hypermedia environment such like WWW.

The task of mining associations between keywords can be stated as follows: Let $K=\{k_1, k_2,\ldots, k_m\}$ be a set of keywords, and let $S=\{s_1,s_2,\ldots,s_n\}$ be a set of structure identifiers in hypermedia documents. In typically speaking, a hypermedia instructional material contains lots of document structures, where each document structure contains a set of keywords.

Example 2. Consider the hypermedia instructional materials shown in TABLE I. (TABLE I will be used as a running example in this section). The keywords in the keyword set *K={O,C,M,I,V}* are "Object", "Class", "Method", "Inheritance", "Variable". Moreover, the document structure set is $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$. To simplify and make it more readable, we use CMV to represent a keyword set, {C, M, V}. Similarly, a document structure set {1, 3, 5} will also write as 135 for the same reason. □

The **support** of the keyword set is how many different

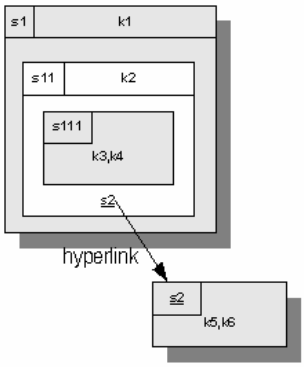


Fig. 2. s_1 and s_2 are structured documents.

document structures that a specific keyword appears on it. A keyword set will be **frequent** if its support value is higher than the expected threshold.

TABLE V.
FREQUENT KEYWORD SETS IN THE STRUCTURED DOCUMENTS WITH *MINSUP*=50%.

FREQUENT KEYWORD SET *MINSUP* >= 50%	SUPPORT
C	100%
V, CV	83%
O, M, I, OC, OV, CM, CI, OCV	67%
OI, MV, IV, OCI, OIV, CMV, CIV, OCIV	50%

A ***confident association rule*** is denoted as $\{k_1, k_2, ..., k_n\} \xrightarrow{p} k_m$, which means that if all of the keywords k_1, k_2, …, k_n can be found out in the specific document's structure, then there should be a good chance to find the keyword k_m out from the same document structure, too. The

acceptance ratio (probability) for such kind of association rules is called the ***confidence*** of the rule. In practical, researchers only care those association rules with high confidence.

Example 3. One extracted rule for Java instructional materials on the Sun website is $\{subclass, supclass\} \xrightarrow{0.85} inheritance$. This rule means that the document structure in the instructional material mentions the "subclass" and "superclass" will also mentions the "inheritance". And this rule's confidence will be 0.85 (85%). □

As *Apriori* uses the ***frequent itemset*** to generate rules, a frequent itemset is an itemset whose occurrence ratio is higher than the threshold. An itemset can be seen as a keyword set (K_i) in the instructional material. Beside the itemset, the ***transaction set*** in *Apriori* can be also represented as the document structure (s_i). By using *Apriori* an association lattice of keywords (based on Example 3) can be built more easily as Fig. 3 shown.

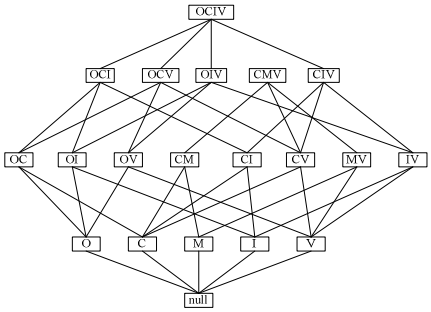


Fig. 3 Association lattice of keywords{O, C, M, I, V} based on Example 3.

In this case there are five keywords (items), {O, C, M, I, V}. The edge in the lattice represents the relation between two keyword-sets (itemsets). The frequent itemset property in Apriori mentioned that any subset of an itemset must be frequent if the parent itemset is frequent. Fig. 4 shows the nonempty subsets of OCV are {OC, OV, CV, O, C, V}. Therefore, according to the frequent itemset property if OCV is frequent, then all of the subsets should be frequent, too.

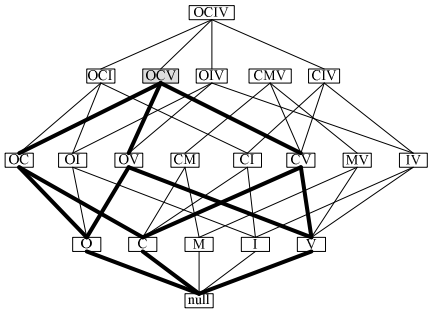


Fig. 4 Subsets of OCV.

After an association lattice of keywords was constructed, the next step is generating confident rules. The step can be divided into three stages:

1. To choose the frequent itemset with a minimal support threshold. For example, when the minimal support =50%:
   C with support 100%;
   V,CV with support 83%;
   O, OC, OV, OCV with support 67%.
2. To find out all the possible rules: (according to the example above)
   $O \rightarrow CV$; $OV \rightarrow C$;
   $OC \rightarrow V$; $V \rightarrow OC$;
   $C \rightarrow OV$; $CV \rightarrow O$.
3. To search the confident association rules with a minimal confidence threshold. For example, when the minconf equals to 100%:
   $O \xrightarrow{1.0} CV$;
   $OC \xrightarrow{1.0} V$;
   $OV \xrightarrow{1.0} C$.

By using the three stages above, the association lattice of keywords can be retrieved automatically For example, if a student wants to study about the topics – {*O, C, V*}, the relevant association lattice of keywords OCV can be discovered as shown in Fig. 5.

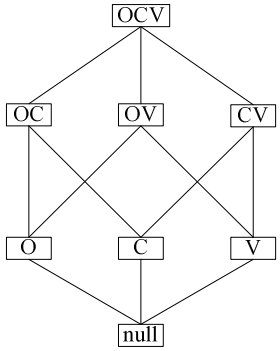


Fig. 5 Association lattice of keywords with a learning topic {O,C,V}.

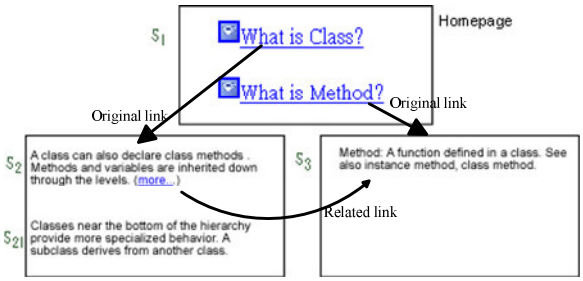


Fig. 6. Re-construct the original webpage and provide related

Example 4. When student reading a courseware of *Java language*. Taking the Sun's Java online lectures for example – http://java.sun.com/docs/books/tutorial/java/index.html, this courseware was writing about the concepts of object-oriented programming. In this document, there is a statement "…A software object implements its behavior with *methods*…". When a student read this statement, he/she might not understand the meaning of the keyword – "methods" clearly. Therefore, the student will need more references for "methods". However, unfortunately, in the most of time, the student will not find out. It is because of the editors of instructional materials never think that will be a question mark in the students' mind. Therefore, if the dependent documents that are associated with the keyword –

"methods" could be retrieved automatically and the original document structure could be re-constructed to more suitable for learning and reading just like Fig. 6 shows, that will be perfect. □

IV. Hypermedia Structure Reconstruction Flow

The hypermedia structure reconstruction flow is divided into three phases. In the first phase, concept association analysis phase, a concept association lattice is built based on the keywords and concept relations (as Fig.7 shows); the second phase, rule extraction phase, discovers association rules from the built concept association lattice (as Fig. 8 shows); and, students then could use the K-Navi toolbar to reconstruct the teaching materials according their requirements at the third phase, structure reconstruction phase (as Fig. 9 shows). After showing the overall picture of the hypermedia structure reconstruction flow, we can reveal the detail operations for each phase.

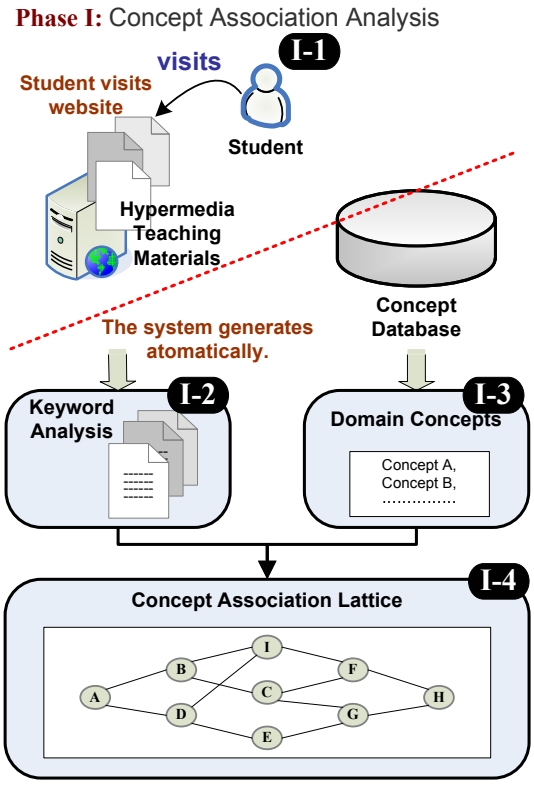


Fig. 7 Keyword analysis and association lattice generation

Phase I is activated by students when they are starting to browse any learning website (step I-1). The system first gathers all connected teaching materials and analyzes the keywords contained in each webpage with pre-defined keyword set (step I-2). Teachers have to define related keyword set for what they plan to teach students in advance. Then, the system retrieves domain concepts from a concept database in order to discover the relations among the teaching materials (step I-3). With the keyword analysis results and domain concepts, the system can build a concept association lattice about the learning website which is visited by students (step I-4, output of phase I).

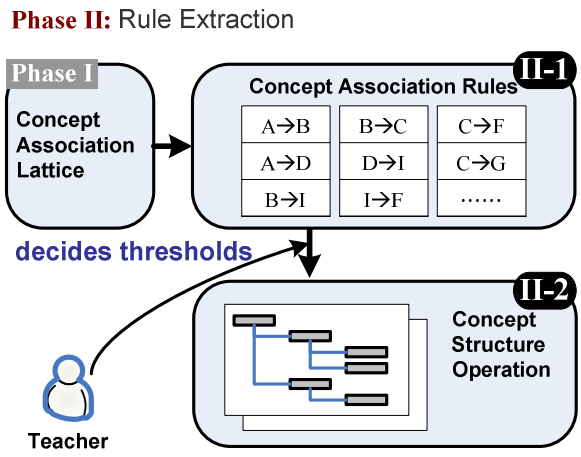


Fig. 8 Association rules extraction

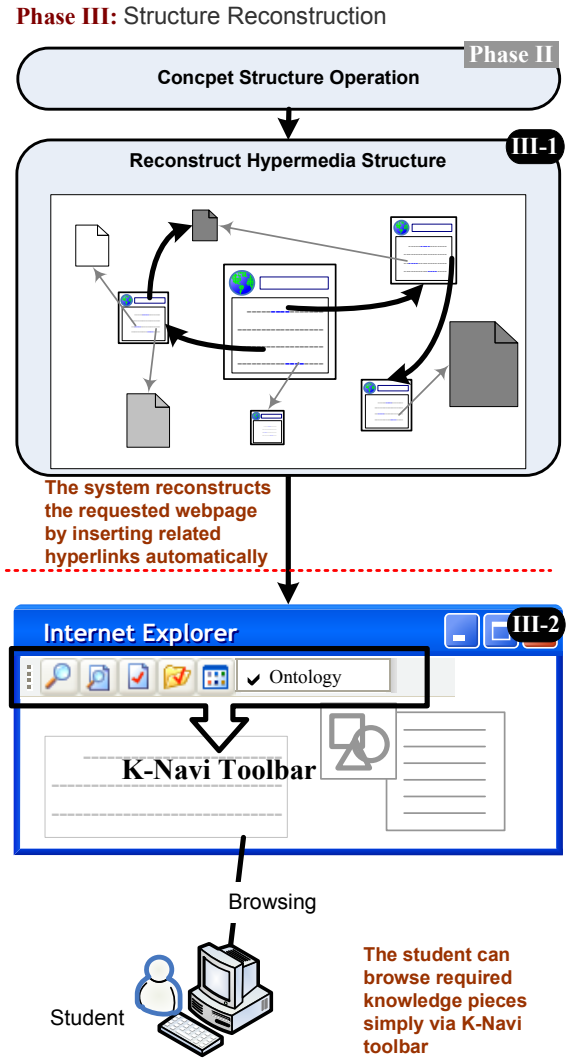


Fig. 9 Reconstructing the hypermedia instructional materials by inserting relevant knowledge pieces' hyperlinks automatically.

Based on the data mining techniques, the system can extract the concept association rules from the output of phase I, the

concept association lattice (step II-1). There might be many rules extracted by the system, therefore, teachers have to define two thresholds, minimum support and minimum confidence, to filter some rules out. The selected association rules then will form a concept structure operation (step II-2, output of phase II).

When the flow comes to the phase III, our goal is almost achieved. The system can reconstruct the hypermedia structure by inserting additional hyperlinks at appropriate position according to the concept structure operation, the output of phase II. (step III-1). As the student studies a topic on the web (step III-2), the K-Navi toolbar will reconstruct the requested webpage which allows the student searches relevant knowledge piece easily and quickly via those auto-generated hyperlinks.

V. K-NAVI TOOLBAR

For realizing the mechanism proposed in this paper, an Internet-based online learning tool, **K-Navi** Toolbar (Knowledge Navigation Toolbar), is implemented as shown in Fig. 10. The K-Navi Toolbar is developed with Borland® C++ Builder 6.0, and the execution environment is built on the Intel® Pentium IV 1.5G MHz CPU + 512MB RAM with Windows® XP Professional and Internet Explorer 6.0.



Fig. 10. Snap-shots of K-Navi Toolbar system.

The K-Navi Toolbar has four major functions as we can see in Fig. 10: 1. keywords markup (【Markup】button); 2. association rule extraction (【Rules Gen】button); 3. document reconstruction (【K-Navi】button); and, 4. visualized association lattice (【FCA】button). We will talk about the details of each function with related snapshots as Fig. 11 to Fig. 15 show.

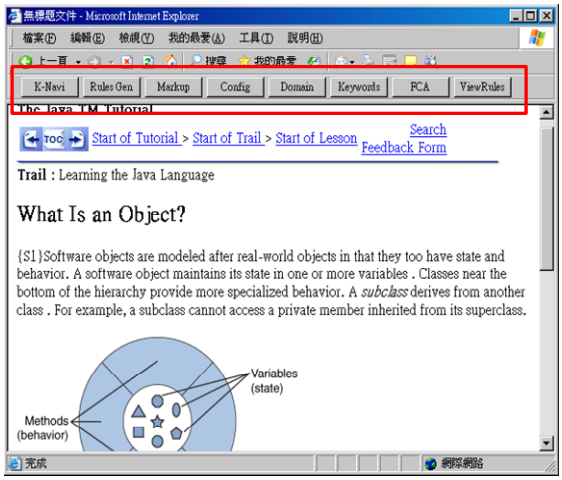


Fig. 11 Reading courseware on the web with **K-Navi Toolbar**.

Fig. 11 shows the experiment framework for a student when he/she reads a teaching material via a web browser. After he/she clicks on the【Keywords】button, a keyword edit window pops up, as Fig. 12 shows.

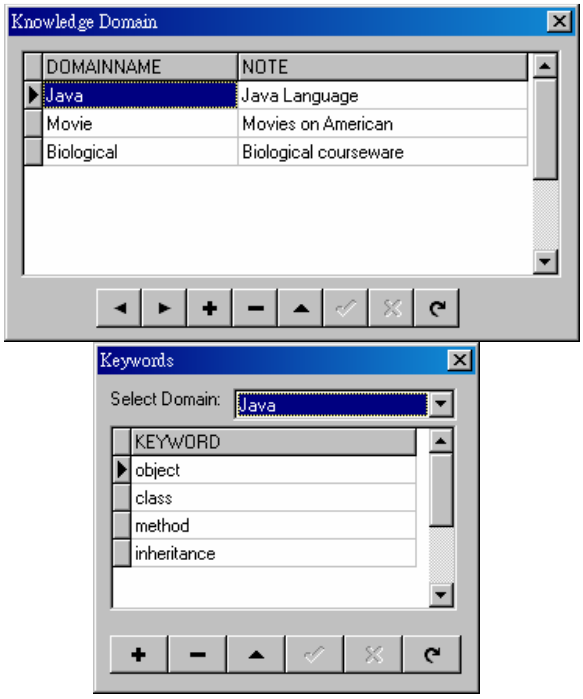


Fig. 12 System configuration.

When the student clicks on 【Markup】button, the system makes the keywords red and bold as Fig. 13 shows. This function can help students get a clear view for knowing which part is important within the teaching material.

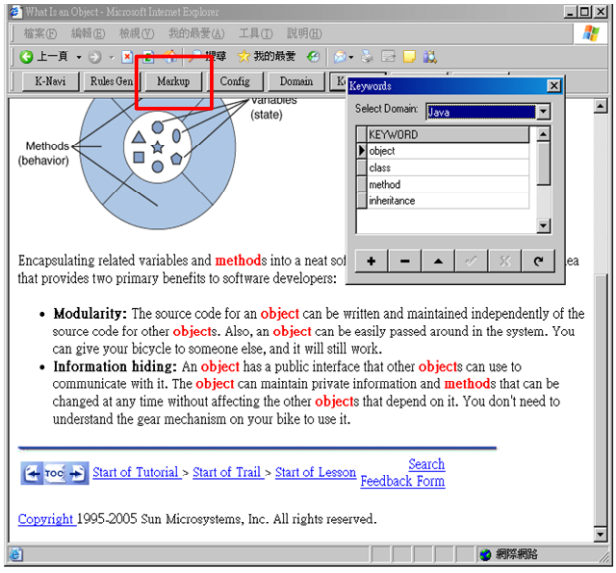


Fig. 13 Keyword markup

The 【FCA】button provides both students and teachers a visualized keyword association lattice. On the right hand side of Figure 14, there is matrix form of keyword relations and possible association rules filtered out according to the support and confidence thresholds.

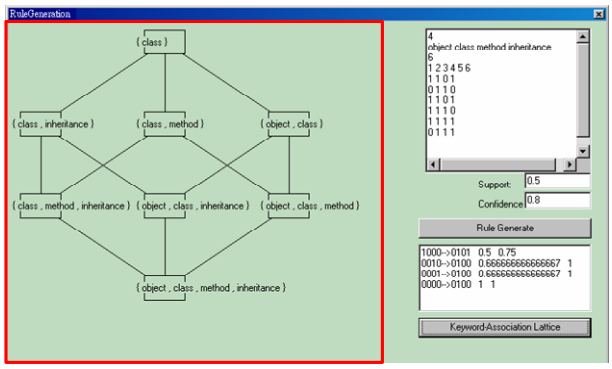


Fig. 14 Keyword Association Lattice

The【Rules Gen】button allows students/or teachers to see the association rules and allows students/or teachers to manage the minimum support and confidence value in order to filter some non-important rules as Fig. 15 shows.

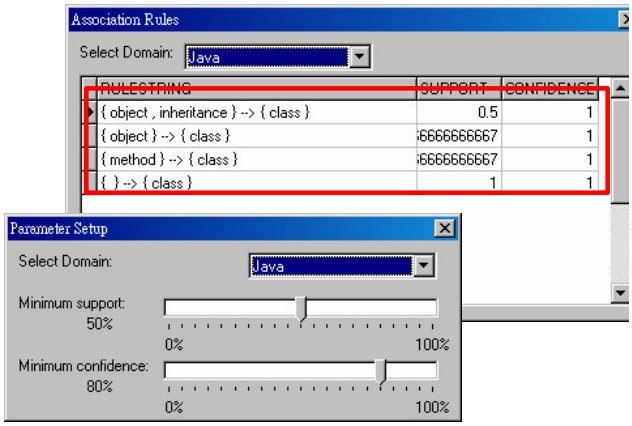


Fig. 15 Association rule extraction

Finally, when students click on the 【K-Navi】button, some relevant knowledge pieces' hyperlinks will be added automatically into the appropriate position of the document just as Fig. 16 shows.

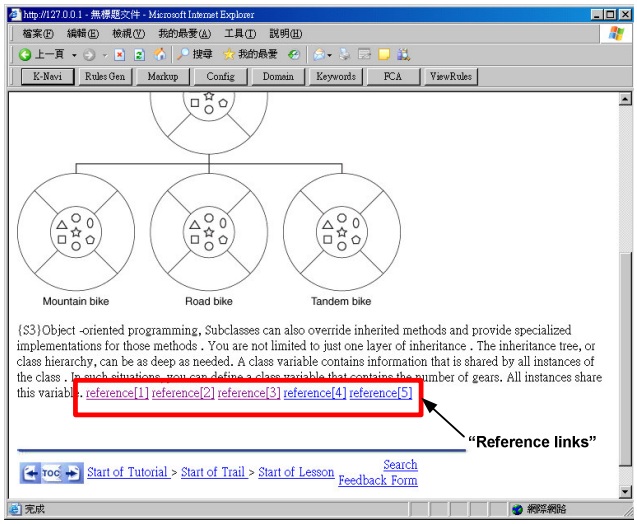


Fig. 16 Additional relevant knowledge pieces' hyperlinks.

This paper applies the K-Navi toolbar to the online object-oriented Java language learning website, Object Oriented Programming of Java language from the Sun tutorial website:

http://java.sun.com/docs/books/tutorial/java/index.html

There are four keywords, including "Object", "Class", "Method", and "Inheritance", within those teaching materials on the learning website.

The contents of the original document structures are shown in Table VI. After parsing the document text, the formal context table can be derived in Table VII.

TABLE VI.
Contents of hypermedia materials with specific keywords.

s_1	Software objects are modeled after real-world objects in that they too have state and behavior. A software object maintains its state in one or more variables. Classes near the bottom of the hierarchy provide more specialized behavior. A *subclass* derives from another class. For example, a subclass cannot access a private member inherited from its superclass
s_2	Class can also declare *class* methods. Methods and variables are inherited down through the levels. In general, the farther down in the hierarchy a class appears, the more specialized its behavior. In addition to instance variables, classes can define class variables.
s_3	Object-oriented programming, Subclasses can also override inherited methods and provide specialized implementations for those methods. You are not limited to just one layer of inheritance. The inheritance tree, or class hierarchy, can be as deep as needed. A class variable contains information that is shared by all instances of the class. In such situations, you can define a class variable that contains the number of gears. All instances share this variable.
s_4	These objects are created when the user launches the application. The application's main method creates an object to represent the entire application, and that object creates others to represent the window, label, and custom component. You can invoke a class method directly from the class, whereas you must invoke instance methods on a particular instance. If one object changes the variable, it changes for all other objects of that type.
s_5	Method: A function defined in a class. See also instance method, class method. Unless specified otherwise, a method is not static. Methods are inherited down through the levels.
s_6	Because the object that represents the spot on the screen is very simple, let's look at its code. The Spot class declares three instance variables: size contains the spot's radius, x contains the spot's current horizontal location, and y contains the spot's current vertical location. It also declares two methods and a constructor — a subroutine used to initialize new objects created from the class. The inheritance tree, or class hierarchy, can be as deep as needed.

TABLE VII
Formal context from TABLE V.

	Object	Class	Method	Inheritance
s_1	1	1	0	1
s_2	0	1	1	0
s_3	1	1	0	1
s_4	1	1	1	0
s_5	1	1	1	1
s_6	0	1	1	1

After applying the minimal support threshold to the lattice, some association rules can be retrieved within its confidence value, as listed in TABLE VIII.

TABLE VIII
Association rules generated by K-Navi system.

Minimal Support	Association Rule	Support	Confidence
0.3	{method}→ {class, inheritance}	0.33	0.5
	{ }→{object, class}	0.667	0.667
	{ }→{class, method}	0.667	0.667
	{ }→{class, inheritance}	0.667	0.667
0.4	{object}→ {class, inheritance}	0.5	0.75
	{method}→{class}	0.667	1
	{inheritance}→{class}	0.667	1
	{ }→{class}	1	1
0.5	{object}→ {class, inheritance}	0.5	0.75
	{method}→{class}	0.667	1
	{inheritance}→{class}	0.667	1
	{ }→{class}	1	1
0.6	**{object}→{class}**	**0.667**	**1**
	{method}→{class}	**0.667**	**1**
	{ }→{class}	**1**	**1**
0.7	{ }→{class}	1	1
0.8	null	null	null

There are three rules when the minimum support is set to 0.6 (60%):

(1){object}→{class};
(2){method}→{class};
(3){ }→{class}.

Based on these three association rules, when a student is reading s_4 (its content is listed in Table IV), the system can recommend the student to read another document in which the keyword "class" is included, because there exist "object" and "method" in s_4. For example, s_2 is a document that talks about the concept "class".

VI. CONCLUSION

A web-based instructional material analyzing and reconstructing system is built in this paper according to the unsupervised reconstruction mechanism. By using keyword association lattice, the relevant hyperlinks between learning resources can be inserted into the original static webpage automatically for individual student. Moreover, the most appropriate instructional materials (webpages) related to what students read can also be able to find out for making suggestions to students. The K-Navi tool was really created to prove that the reconstruction mechanism is workable and reasonable association rules were also discovered. It is worth to note that the experiment system is not only can be used for e-Learning system, but also will be available for any kind of web-based browsing environment.

In the future, we will explore the time performance of generating association rules when the concept association lattice is very large. There are also other possible directions to extend this work:

(1) The keyword database may be able to construct automatically or semi-automatically, thus students could read any on-line materials more conveniently;

(2) The mechanism of extracting association rules from a concept association lattice needs a pre-defined minimum support and confidence threshold. We hope the system could find the most appropriate values for these two thresholds by itself;

(3) The concept association lattice and related association rules should be able to build and extract in real time without requiring students to click on any button to start the process.

REFERENCES

[1] C. Süß, B. Freitag and P. Brssler, "Metamodeling for Web-Based Teachware Management," Proc. Int'l WWWCM99 Workshop on World-Wide Web and Conceptual Modeling, pp.360 – 373, Paris, France, Nov. 15-18 1999.

[2] G. A. Novak, Virtual Courseware for Geoscience Education: Virtual Earthquake and Virtual Dating, *Computers & Geosciences*, Vol. 25, No. 4, pp. 475-488, 1999.

[3] J. Phelps and R. Reynolds, Formative Evaluation of a Web-based Course in Meteorology, *Computers & Education*, Vol. 32, No. 2, pp. 181-193, 1999.

[4] J. X. Chen , The Current Trend of e-Learning Marketing on American, Trans. of I.T. Application for Management, Vol. 24, No. 3, pp.92-96, 2001.

[5] M. Rosenberg, *E-Learning: Strategies for Delivering Knowledge in the Digital Age*, MacGraw-Hill Companies, Inc., 2001.

[6] P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab, "Learning taxonomic relations from heterogeneous sources," Proc. of the ECAI 2004 Ontology Learning and Population Workshop, 2004.

[7] P. Cimiano, S. Staab, and J. Tane, "Deriving concept hierarchies from text by smooth formal concept analysis," Proc. of the GI Workshop (LLWA), pp. 72-79, 2003.

[8] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, chapter 12, pp. 307-328. AAAI Press, 1996.

[9] R. Godin and R. Missaoui, "An incremental concept formation approach for learning from databases," Theoretical Computer Science, Vol.133, pp.387-419, 1994.

[10] T. Berners-Lee, Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor, New York: HarperCollins Publishers Inc., 1999

[11] T. Chang, "HieNet: A user-centered approach for automatic link generation," Proc. of the ACM Conference on Hypertext (Hypertext'93), Seattle, WA, USA, pages 145-158, November 1993.