

Integrating Concept Mapping into Designing a Course Management System

HSIN-YIH SHYU
Tamkang University
Taiwan
hyshyu@mail.tku.edu.tw

SHANG-HSIEN HSIEH
National Taiwan University
Taiwan
shhsieh@ce.ntu.edu.tw

YU-HUR CHOU
Tung-Nan Institute of Technology
Taiwan
yhchou@mail.tnit.edu.tw

In higher education, different instructors who teach the same course may vary the instructional paths, pace, and depth of the content. These variations may result in different performances among learners. The intent of this research is to design a course management system that aggregates appropriate course units based on maximum consensus among instructors, and to distribute contents to the learners more coherently and effectively. The concept mapping method is used as the model for the aggregation of course units, the construction of course structures, and the establishment of learning paths. Based on Goldsmith's closeness index (Goldsmith, Johnson & Acton, 1991), a new algorithm to integrate different course structures is also proposed. In order to explain the mechanism and functions that have been developed for this system, a civil engineering course (i.e., Reinforced Concrete Design) is used as an example. This study is significant since it provides

a new algorithm for aggregating learning objects in course-management systems.

Traditionally, the programs and curricula in higher education have been divided into several pre-specified courses. Each of these courses is delivered over a fixed period and involves a relatively stable learning path. However, different instructors may, while teaching the same course, vary the instructional paths, content, pace, and depth of content. These variations may result in learners performing differently. This method of teaching, with only minor changes, has been constant for several decades. Today, the advances in e-learning technology have brought about a learning paradigm shift in the way we learn and the methods we use to teach and deliver knowledge. With e-learning technology, learning can be distributed such that the right content is provided for the right student at the right time. Thus, the traditional teacher-centered course has been moving toward a learner-centered one in order to provide adaptive and appropriate learning materials and modes for students with different needs.

A course management system (CMS) is a software package designed to organize learning objects in order to help educators create quality courses. In order to use a more rational decision making process, an appropriate technique and algorithm that is capable of managing those objects becomes very important. This study used concept mapping as a technique to design the course management system. The concept map is a technique used to represent knowledge using graphs. Knowledge graphs are networks of concepts; these networks consist of nodes (points/vertices) and links (arcs/edges). Nodes represent concepts and links represent the relationships between such concepts. The concept mapping technique depicts a graphical node-arc representation of the relationships among a collection of propositions. Each proposition that constructs a semantic unit consists of a set of concept nodes that are interconnected with each other by means of relational links. Gupta (2000) suggested that there are several advantages in using the approach of concept mapping techniques in managing instruction and learning. These include the following: (1) a concept map brings out clarity in logical development of the theme; (2) the recognition of interdependence among concepts helps in identifying crucial steps in understanding the theme; (3) a concept map provides a hierarchical overview of the domain knowledge; (4) a concept map helps the instructors to recognize any missing links among the concepts in any body of knowledge; (5) a concept map could provide a

structure for development of design procedures and development of educational materials. In addition, there are certain applications of concept mapping techniques in education, such as (1) the representation of knowledge (Anderson, 1995; Turns, Atman, & Adams, 2000), (2) the assessment and diagnosis of the learning effect, and (3) the analysis of the learning path. Many studies (Fung & Kemp, 2000; Lee & Liu, 2002; Lin & Chen, 2002; Reyes & Sison, 2000) have discussed the learning path and the course management system, but few of them use the concept mapping technique. In this paper, the concept mapping technique is implemented as a model for aggregating course units, constructing course structures, as well as establishing a learning path. In addition, a new algorithm to integrate different course structures, based on Goldsmith's closeness index, is also proposed. In order to explain the mechanism and functions that have been developed for this system, a civil engineering course (i.e., Reinforced Concrete Design) is used as an example to conduct a formative evaluation for this system. Finally, the course management system is presented at the end of the paper.

SYSTEM ANALYSIS AND DESIGN

The proposed system design diagram for the course management system (abbreviated as CMS) is presented in Figure 1. The various conceptual elements designated by the ellipses and cylinders in the diagram represent the attributes for each course unit. There are seven attributes, including the contents, references, examples, tests, evaluated parameters, and link information for front course units and rear course units. Among them, only the contents, examples, and tests are considered to be learning objects because of their reusability and interoperability. The design procedures of this course management system can be sequentially divided into four steps: the definition of course units, the construction of sub-nodes, the construction of nodes, and the construction of chapters. They are illustrated in detail as follows:

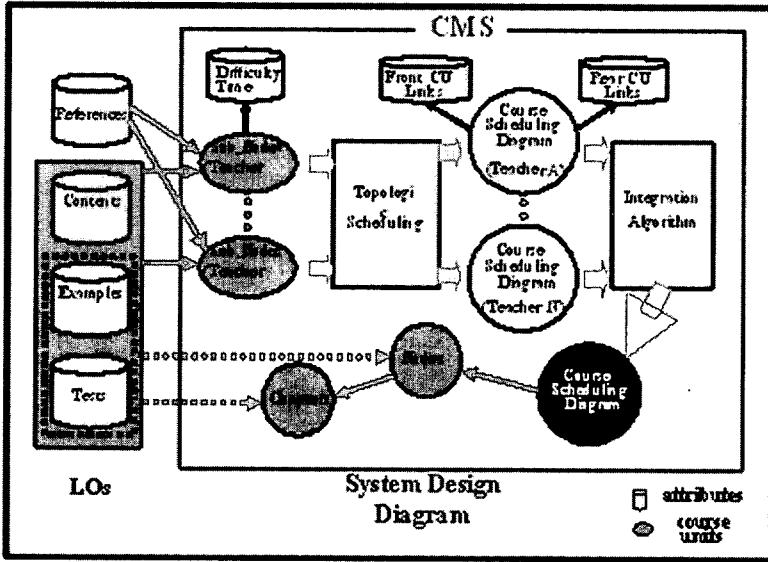


Figure 1. System diagram of the course management system

The Definition of the Course Unit

A course structure is separated into three levels and is presented in Figure 2. The course unit has three classes, namely, chapters, nodes, and sub-nodes. Gupta (2000) introduced the idea of a concept map to represent the course units and claimed that a concept mapping technique is a well-defined step in the development of the domain knowledge. In this study, the concept mapping technique is used to represent a course unit for each class.

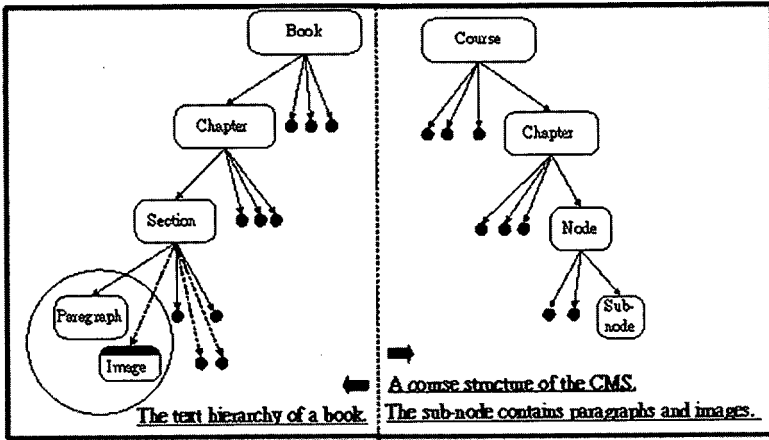


Figure 2. An explanation of a course structure

I. The concept map of a course unit (sub-node)

The diagram of a sub-node is shown in Figure 3. Each sub-node contains seven attributes, which are described as follows:

- (1) **Contents:** The principal part of the sub-node, which includes teaching purposes, principles, and materials.
- (2) **References:** The relational links of the sub-node. Each reference holds a weighted parameter representing the important factor in the link.
- (3) **Examples:** Examples, by their design, to explain the teaching principles.
- (4) **Tests:** Most of the tests comprise criterion-referenced tests and contain diagnosis parameters to evaluate students' misconceptions.
- (5) **Evaluated parameters:** Parameters, such as difficulty level and time-on-task, used to evaluate the learning path and learning tactic.
- (6) **Front course units:** Linking information regarding pre-course units that contain weighted parameters expressing the important factors in the front course units.
- (7) **Rear course units:** Linking information regarding rear-course units that contain weighted parameters expressing the important factors in the rear course units.

II. The concept map of a course unit (node)

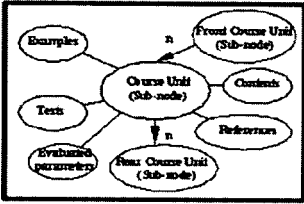


Figure 3. The concept map of a course unit (sub-node)

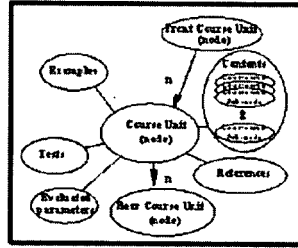


Figure 4. The concept map of a course unit (node)

The concept map of the node is shown in Figure 4, and is similar to the sub-node that contains seven attributes. The intention of the node's attribute is comparative integration. Tests within the node comprise norm-referenced tests, and examples of the node are used to explain a series of theorems.

III. The concept of a course unit (chapter)

The concept map of the chapter is similar to the concept map of the node. The collection of chapters comprises all of the contents of a course. There are several advantages in using concept mapping to construct the course structure. Since a course structure is broken down into a tree-branching diagram, teachers may easily identify and understand the relationships among the different course units, and may reassign and design different learning paths for learners using different teaching strategies. The evidence has shown that using concept mapping improves the flexibility of the course development (Gupta, 2000; Turns, Atman, & Adams, 2000).

The Construction of the Course Unit (Sub-node)

The first step is to collect sub-nodes from different textbooks and make those sub-nodes that cover the same topic unique. If different sub-nodes cover the overlapped topic, the construction of course structure will encounter some problems. As an example, the course structure will not conform to a tree structure and the relation between course units will be in conflict. Each sub-node contains seven attributes as shown in Figure 1. The evaluated parameters of each sub-node include difficulty ($UU_i(D)$) and time-on-learn-

ing $(UU_i(T))$. The equations are used to integrate evaluated parameters for different assignments from teachers, as follows:

$$UU_i(D) = \frac{\sum_{j=1}^n (UU_{ij}(D) * W_j)}{\sum_{j=1}^n W_j} \text{----- (1)}$$

$$UU_i(T) = \frac{\sum_{j=1}^n (UU_{ij}(T) * W_j)}{\sum_{j=1}^n W_j} \text{----- (2)}$$

W_j : the weighted value for teacher i.

$UU_{ij}(D)$: the difficulty of course unit I (Teacher j assigns).

$UU_{ij}(T)$: the time on learning of course unit I (Teacher j assigns).

The above information is valuable for instructional design.

The Construction of the Course Unit (Node)

The node is a collection of selected sub-nodes. The difficulty of the node is the average of the selected sub-nodes and the time-on-learning is the sum of the selected sub-nodes. They are defined as follows:

$$U_i(T) = \sum_{j=1}^n UU_j(T) \text{----- (3)}$$

$$U_i(D) = \frac{\sum_{j=1}^n UU_j(D)}{n} \text{----- (4)}$$

Relations between Course Units

Each sub-node needs to be assigned the rear and front sub-nodes and searches for the timetable. The method of searching for the course timetable is described in the following section. There are four relationships between course units at the same level. These relationships are expressed in Table 1.

Table 1
Relations between course units

Name	Code Name	Illustration
Independent Relation	Ind	No relation
And Relation	And	All units are finished, and then go to next.
Or Relation	Or	Any unit is finished, and then go to next.
Ordered Relation	Order	The execution must follow the order.

The topologic scheduling begins when the relations' assignment is finished. The following example in Table 2 is used to explain the topologic scheduling. Sub-node B must be executed after sub-node A has finished. Topologic scheduling must wait until both sub-nodes (I and J) are finished, before going to next sub-node.

Table 2
Relations between units (Example)

Sub-node	Related Sub-node	Relation	Sub-node	Related Sub-node	Relation
A	B		B	C	
I	J	And	H	K	And
C	D,E		D	F	
E	H		F	I	
K	G		I	G	

A course sequence diagram, shown in Figure 5, is found after executing the topologic scheduling using the information in Table 2:

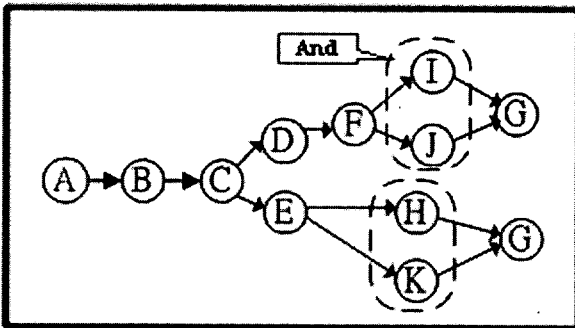


Figure 5. A course sequence diagram

Typical learning time and difficulty regarding a node will be calculated, if the data for all sub-nodes are already known. The time-on-learning and the difficulty in relation to this example are shown in Table 3. The statistical table is shown in Table 4.

Table 3

The difficulty and time-on-learning of the sub-node (Example)

Sub-node	A	B	C	D	E	F	G	H	I	J	K
Difficulty	0.2	0.3	0.4	0.2	0.5	0.7	0.2	0.3	0.6	0.3	0.4
Time (h)	2	1	1	1	2	2	2	3	1	1	3

The time-on-learning is calculated and the difficulty of a learning process is verified from the beginning to a specified sub-node by using the data in Table 3 and the sequence diagram (in Figure 5). The equations used to perform the calculations are as follows:

Learning time of a learning process = the sum of time on learning (all front sub-nodes)

Average difficulty of a learning process = the average difficulty (all front sub-nodes)

Absolute difficulty of a learning process = the one with the maximum difficulty (all front sub-nodes)

The statistical results, based on the above equations, are shown in Table 4

Table 4

Time on learning and difficulty of a learning process

Process to sub-node	A	B	C	D	E	F	G		H	I	J	K
							Up	Down				
Absolute difficulty	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.5	0.5	0.7	0.7	0.5
Average difficulty	0.2	.25	0.3	.28	.35	.36	.36	.33	.34	.4	.35	.36
Time on learning (h)	2	3	4	5	6	7	11	14	9	8	8	9

Using the information in Table 4, the learning process statistic is calculated and the shortest learning path identified. For example, the shortest period of time to be spent on learning is 11 hours for the learning process that extends from the beginning to the sub-node G.

The Integration of Different Learning Paths

Since different instructors may, while teaching the same course, vary the instructional paths, content, pace, and depth, this section proposes a mechanism to integrate different teaching schedules. This mechanism is described, using the following example, in the subsections below.

Four teachers construct four different sequences for the same course. The four patterns of sequences are shown in Figure 6.

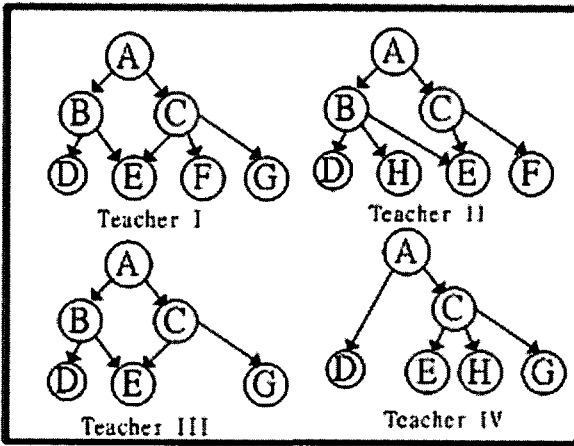


Figure 6. Four different course sequences

I. Compute the confidence values of teachers by using the closeness index.

Goldsmith’s method (Goldsmith, Johnson, Acton, 1991; Chen, Lin, & Chang, 2001) is used to compute the closeness index and the computing processes. The steps involved are listed as follows:

- Determine all sub-nodes $N = N1 \cup N2$.
- For each sub-node (n_i) belonging to N
 Locate its first-order neighbor sets.
 Compute the intersection, $I^{(i)}$ and the union, $U^{(i)}$.
 Calculate the closeness coefficient, $C^{(i)} = I^{(i)} / U^{(i)}$.
- Compute the closeness index, $C(H1,H2)=1/N\sum C^{(i)}$. The value of $C(H1,H2)$ represents the closeness index.

A comparison between teacher (H1) and teacher (H4) is made. The closeness index is shown in Table 5. Based on the closeness index algorithm, the results of the comparison among the four teachers are shown in Table 6. Each teacher obtains a confidence value.

Table 5
Results of comparison between two teachers

I	H1	H4	$I^{(0)}$	$U^{(0)}$	$C^{(0)}$
A	{B,C}	{D,C}	{C}	{B,C,D}	0.333
B	{A,D,E}	{}	{}	{A,D,E}	0
C	{A,E,F,G}	{A,E,H,G}	{A,E,G}	{A,E,F,H,G}	0.6
D	{B}	{A}	{}	{A,B}	0
E	{B,C}	{C}	{C}	{B,C}	0.5
F	{C}	{}	{}	{C}	0
G	{C}	{C}	{C}	{C}	1
H	{}	{C}	{}	{C}	0

The closeness index $C(H1,H4) = 0.304$

Table 6
Confidence Values

Closeness Index	Teacher I	Teacher II	Teacher III	Teacher IV
Teacher I		0.688	0.821	0.304
Teacher II	0.688		0.531	0.154
Teacher III	0.821	0.531		0.369
Teacher IV	0.304	0.154	0.369	
Total	1.813	1.373	1.721	0.827
Confidence Value	0.316	0.240	0.3	0.144

II. Integration of course sequences

In this section, the integrated algorithm for integrating different course sequences is discussed; the procedures of the integrated algorithm are explained in the following two steps:

Step1: Using a union algorithm, the system expands a course sequence for each teacher, and constructs a unique course sequence. The extended course

sequences are shown in Figure 7. The weighted values of the link and the unit are calculated as follows:

- (1) The weighted values of new units (links) for each course sequence are zero.
- (2) The weighted values of original units (links) for each course sequence are the confidence value of the teacher.
- (3) Accumulate the weighted value for each unit (link). If the sum of the weighted values is greater than 0.5, the connected units (links) are kept.

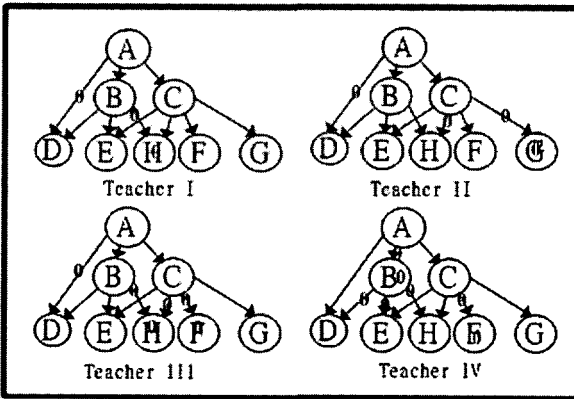


Figure 7. Extended course sequences

Step 2: After processing the integration of course sequences, the weighted values for the course units are shown in Table 7. The weighted values for the links are shown in Table 8.

Table 7
The weighted values of course units

Unit Teacher	A	B	C	D	E	F	G	H
I (.316)	1	1	1	1	1	1	1	0
II (.24)	1	1	1	1	1	1	0	1
III (.3)	1	1	1	1	1	0	1	0
IV (.144)	1	0	1	1	1	0	1	0
Total	1	.856	1	1	1	.556	.76	.24

Table 8
The weighted values of links

Teacher \ Link	AB	AC	AD	BD	BE	BH	CE	CH	CF	CG
I (.316)	1	1	0	1	1	0	1	0	1	1
II (.24)	1	1	0	1	0	1	1	0	1	0
III (.3)	1	1	0	1	1	0	1	0	0	1
IV (.144)	0	1	1	0	0	0	1	1	0	1
Total	.86	1	.14	.86	.62	.24	1	.14	.56	.76

In Table 7 and Table 8, the total value is used as the sieve. Units and links whose total value is less than 0.5 are deleted. From the data results of Table 7 and Table 8, seven units (A, B, C, D, E, F, and G) and seven links (AB, AC, BD, BE, CE, CF, and CG) are retained. The integrated course sequence is shown in Figure 8:

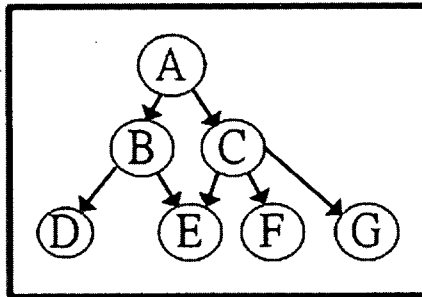


Figure 8. The integrated course sequence

III. The establishment of the node unit

So far, only the links of sub-nodes in the same node have been discussed, since the links of sub-nodes between different nodes are used as references and are irrelevant to controlling the learning path. This may be explained using the following example. A chapter is divided into 26 sub-nodes (A~Z). First, by using the topological algorithm, the relationships between sub-nodes and searches for the course sequence are found. The course sequence, in Figure 9, is the result obtained once topological searching has taken place. Second, sub-nodes of the aggregate node are selected. This chapter is divided into four nodes; the information regarding the selected sub-nodes for each node is presented in Table 9.

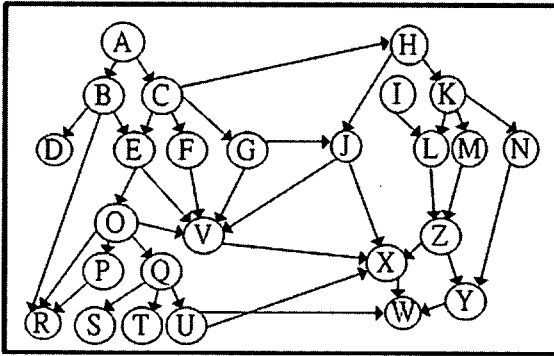


Figure 9. Course sequence

Table 9
Selected sub-nodes of node

Node number	Selected sub-nodes
1	A, B, C, D, E, F, G
2	H, I, J, K, L, M, N
3	O, P, Q, R, S, T, U, V
4	W, X, Y, Z

After aggregation, the shape of each node is shown in Figure 10. The links that are expressed as dotted lines between nodes are references. The links within each node are used as the learning path control.

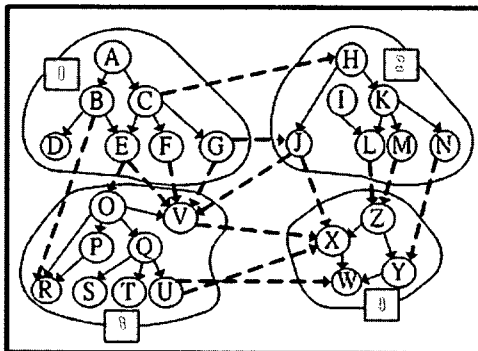


Figure 10. The course sequence between nodes

At the sub-node level, all links are regarded as references. For example, the references of sub-node J are G, H, V, and X. At the node level, node 1 has two references (node 2 and node 3).

The Construction of the Course Unit (Chapter)

The chapter is composed of nodes. From the above example, the course sequence of the chapter is shown in Figure 11. The logic of generation is the same as in the case of the construction of the node.

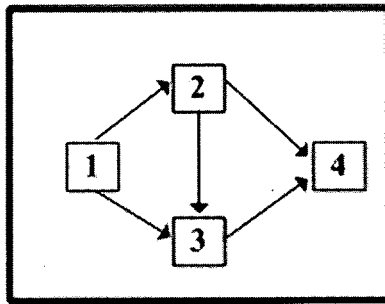


Figure 11. The course unit (Chapter)

CASE STUDY: EXAMPLE

In this section, we use a textbook on civil engineering (i.e., reinforced concrete design) written by Nawy (1996) as the example to test the design procedures for this course management system. The testing procedures are explained as follows:

Splitting a Course into Sub-Nodes According to the Teaching Strategy

To test this system, four instructors who teach Reinforced Concrete Design at the undergraduate level, were invited to provide formative evaluation of the system. They were advised to consider the student's background, the learning environment, the teaching method, and the teaching objective in order to split the course. The analyses of the course used in the example are described as follows:

Objective: Teaching students how to design reinforced-concrete structures.

Student’s background: Sophomore, first-time learner, and a conceptual understanding of structural mechanics.

Teaching method: Project-based learning using a reinforced-concrete structure design project for teaching.

Teaching mode: Web-based course, with mid-term and final exams using written tests

Table 10 shows the partial sub-nodes, which have integrated those four instructors’ course structures.

Table 10
Partial Sub-nodes for this exemplary course

Sub-node	Code	Difficulty (0~1)	Time (Min.)
Concrete Material	A	0.2	30
Ductility and Plasticity	B	0.3	15
Design Strength	C	0.1	10
Required Strength	D	0.1	10
Reinforced Bar	E	0.2	20
The Modulus of Elastic (Steel)	F	0.2	10
Clear Span Length and Safety Layer	G	0.4	40
Concrete Compressive Strength	H	0.2	10
The Modulus of Elastic (Concrete)	I	0.2	10
Concrete Tensile Strength	J	0.3	15
Safety Rules	K	0.4	20
Basic Assumptions	L	0.4	20
Single Reinforced Section Strength Analysis (Rectangle)	M	0.5	40
Double Reinforced Section Strength Analysis (Rectangle)	N	0.6	40
Single Reinforced Section Strength Analysis (T Section)	O	0.6	40
Double Reinforced Section Strength Analysis (T Section)	P	0.7	40

Constructing the Relational List of Sub-Nodes

The relational list for the exemplary course is shown in Table 11.

Table 11
 Relational list for the exemplary course

Sub-Node	Related Sub-node	Relation	Sub-Node	Related Sub-node	Relation
C	D	And	E	F, G	And
A	H, I, J	And	L	K	And
L	O		C	K	
B	A, E		M	N	
O	P		A	M, O	
E	M				

Preparing the Topological Scheduling Diagram

The scheduling diagram for this exemplary course, which is presented in Figure 12, uses the relational data in Table 11 and the information regarding the sub-nodes in Table 9.

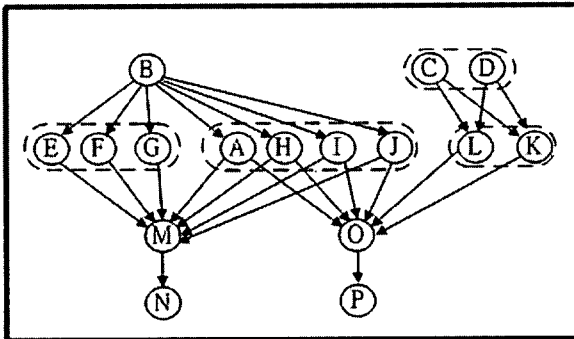


Figure 12. Scheduling diagram of sub-nodes

Integration of Different Course Sequences

When teachers produce different scheduling diagrams, the use of the integrated algorithm is recommended.

Developing Nodes

The sub-nodes for aggregating nodes are selected. The selected sub-nodes for each node are shown in Table 12.

Table 12
List of selected sub-nodes

Node Number	Node Name	Selected Sub-nodes
1	Material Properties	A, B, E, F, G, H, I, J
2	Basic Specifications	C, D, K, L
3	Beam Strength Analysis (Rectangle Section)	M, N
4	Beam Strength Analysis (T Section)	O, P

The scheduling diagram of the nodes, as it appears after combining selected sub-nodes (Table 12) with the scheduling diagram of sub-nodes for the exemplary course, is shown in Figure 13:

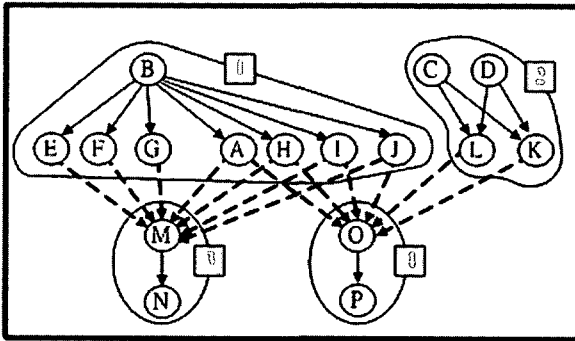


Figure 13. The scheduling diagram of nodes

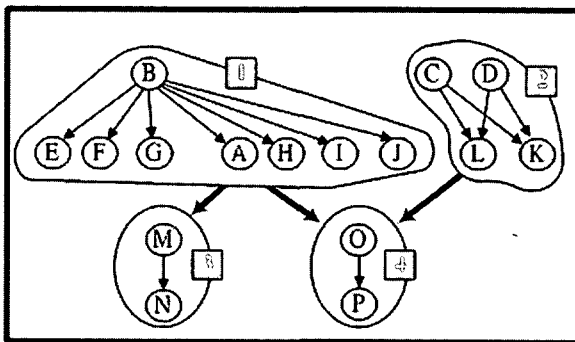


Figure 14. The scheduling diagram of the exemplary course

Based on the results presented in Figure 13, the scheduling diagram of the exemplary course is produced; this is shown in Figure 14. To summarize, the following two results are used to depict the learning path control for the adaptive learning system.

- The diagram of the course structure includes information regarding levels and ordering.
- The attribute information regarding course units for different levels includes contents, references, tests, examples, evaluated parameters, front units, and rear units.
- The attribute information regarding course units for different levels includes contents, references, tests, examples, evaluated parameters, front units, and rear units.

THE COURSE MANAGEMENT SYSTEM

Based on the algorithms described above, a course management system (CMS) was developed; this is shown in Figures 15 to 20. VB™ software is used as the programming language and ACCESS™ as the database system. Users may click one of the buttons at the bottom of the screen to construct the sub-nodes, nodes, or chapters on the entry screen (shown in Figure 15). The procedure for building a course structure follows a bottom-up algorithm. It is suggested that the teacher develop the sub-nodes first, and then construct the nodes and chapters last. Figure 16 shows how a sub-node can be constructed by selecting learning objects including content, examples, and tests:

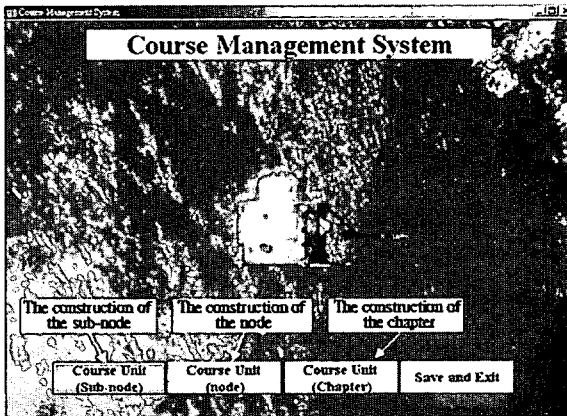


Figure 15. The entry screen of the CMS

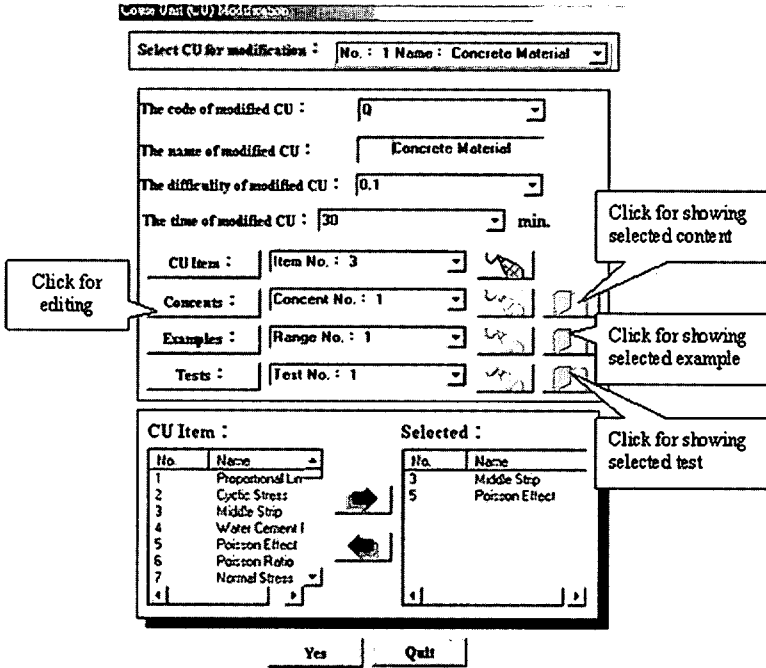


Figure 16. Construction of a sub-node

The teacher can create relationships among sub-nodes by means of easy clicks, as shown in Figure 17. After the execution of the topological algorithm, the system displays the statistical results of scheduling (Figure 18) and the scheduling diagram of a course (Figure 19). Figure 20 illustrates the function of the integration of different course sequences. There are several advantages to using this CMS. It is easy to operate (requiring only clicks apart from typing), and is both automatic and adaptive. The CMS provides a useful tool for course developers as well as teachers to construct a course structure and manage course units.

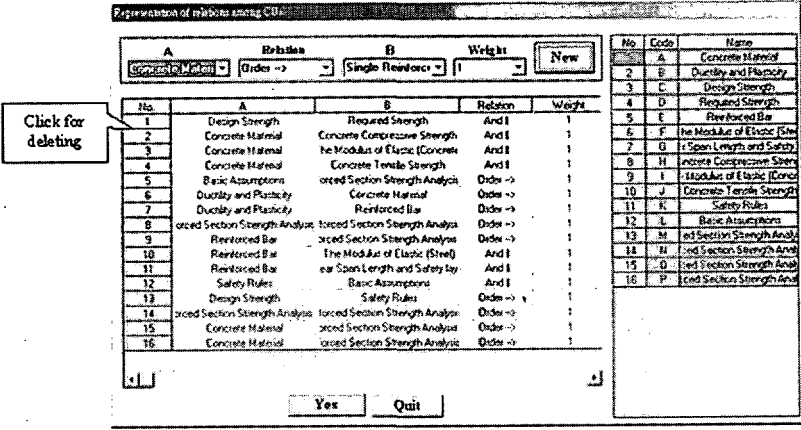


Figure 17. Relations between course units

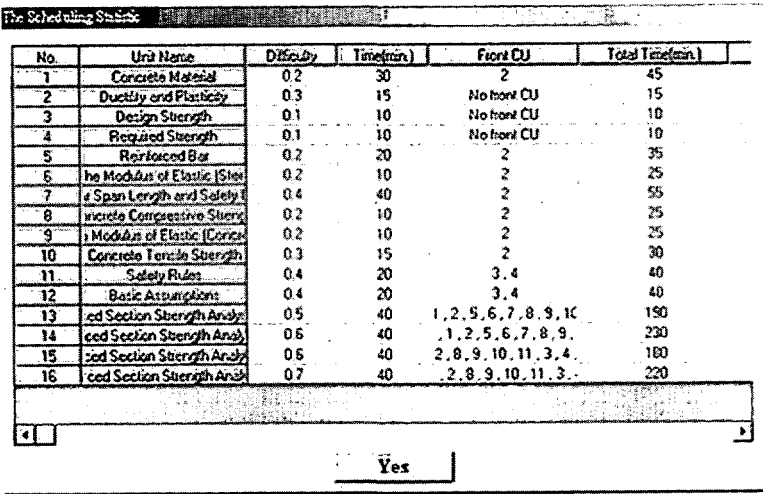


Figure 18. The statistical results of scheduling

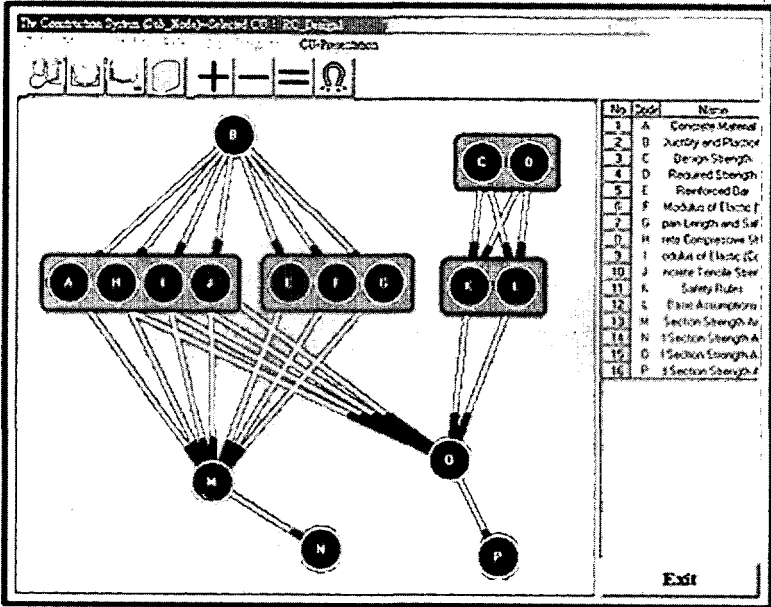


Figure 19. Scheduling diagram of a course

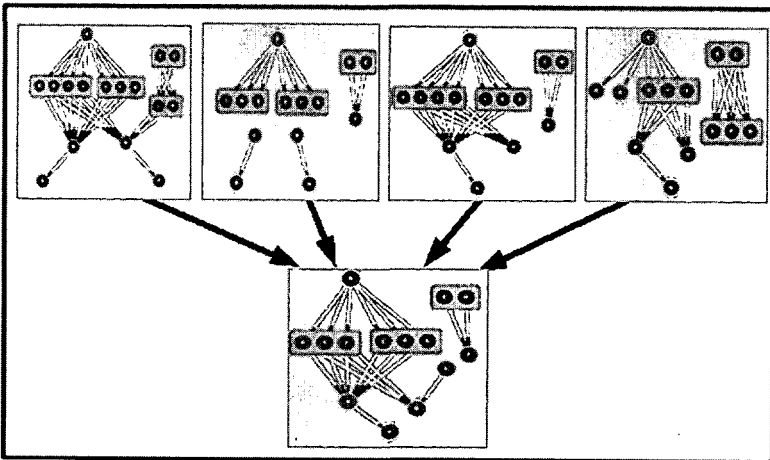


Figure 20. Integration of four different instructors' course sequences

CONCLUSION

In this paper, a new algorithm for course design based on the concept mapping technique is proposed. A course is separated into course units involving three levels that include the sub-node, node, and chapter. The technique of concept mapping is used to represent a course unit for each level, and a tree-branching and hierarchical course structure is constructed. This algorithm is implemented to integrate different course structures as well as to provide useful parameters for managing the learning paths. The algorithm is allowed to produce scheduling diagrams that can clearly identify relationships among different course units, and help teachers discover any missing course units or links. Thus, the course management system involving this algorithm is more flexible relative to aggregating learning objects. This study is significant since it provides e-learning curriculum developers with a new algorithm for designing a course management system.

References

- Anderson, J. R. (1995). *Cognitive Psychology and its Implication*. New York: Freeman and Company.
- Chen, S., Lin, S., & Chang, K. (2001). Attributed Concept Maps: Fuzzy Integration and Assessment. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 31(5), 842-852.
- Cien, L. U., Lin, S. Y. & Wu, Y. L. (2001). Conceptual Graph for Assisted Learning on the Web. *Proceedings of the 5th Global Chinese Conference on Computers in Education*, 2001.
- Fung, P. & Kemp, R. (2000). Traversing the Case Graphs: A Computer Model for Developing Case-based Learning Systems. *Proceedings of ICCE 2000*, Vol. 2, 906-913, Nov. 21-24 Taipei.
- Goldsmith, T., Johnson, P., & Acton, W. (1991). Assessing structural knowledge. *Journal of Educational Psychology*, 86, 88-96.
- Gupta, K. C. (2000). Concept Maps and Modules for Microwave Education. *IEEE Microwave Magazine*, 1(3), 56-63.
- Lee, M., & Liu, Y. S. (2002). Applying Neural Networks to Integrate Concept Maps to Build Web-based Courseware. *Proceedings of the 6th Global Chinese Conference on Computers in Education*, Beijing.
- Lin, M. H. & Chen, H. R. (2002). An Authoring Tool of Instruction Outlines and Materials on the Internet. *Proceedings of the 6th Global Chinese Conference on Computers in Education*, Beijing.
- Nawy, Ed (1996). *Reinforced Concrete: A Fundamental Approach*. Englewood Cliffs, New Jersey: Prentice Hall.

- Reyes, R. L. & Sison, R. (2000). CBR-TUTOR: A Case-Based Reasoning Approach to an Internet Agent-based Tutoring System. *Proceedings of ICCE 2000*, Vol. 2, 1413-1420, Nov. 21-24, Taipei.
- Turns, J., Atman, C. & Adams, R. (2000). Concept Maps for Engineering Education: A Cognitively Motivated Tool Supporting Varied Assessment Functions. *IEEE Trans. on Education*, 43(2), 164-173.

Acknowledgements

The authors would like to thank the National Science Council, R.O.C. (Taiwan) for funding this study. The grant number is NSC-90-2516-S-032-001.

A vertical bar on the left side of the page, consisting of a series of horizontal segments in shades of yellow and orange, with a small red diamond at the top.

COPYRIGHT INFORMATION

TITLE: Integrating Concept Mapping into Designing a Course
Management System

SOURCE: J Educ Multimedia Hypermedia 13 no4 2004

WN: 0400405878010

The magazine publisher is the copyright holder of this article and it is reproduced with permission. Further reproduction of this article in violation of the copyright is prohibited.

Copyright 1982-2005 The H.W. Wilson Company. All rights reserved.