Sci. J. Fac. Sci. Menoufia Univ. Vol. VIII (1994)

INTELLIGENT COURSEWARE SYSTEM

Dr. H. M. Harb
Azhar University, Faculty of Engineering,
Computers and Systems Engineering Department

ABSTRACT

This paper presents an intelligent courseware system. The system is composed of four submodels:the courseware submodel, the user submodel, the courseware text and graphics editor and the presenter.It satisfies the instructor and student requirements. The instructor as a system administrator can create student accounts, edit courseware and student grades. The system has an authentication subsystem allowing a secure login to the system. The courseware is divided into units forming a hierarchy and the student can consult a unit any number of times. Each unit is followed by an arbitrary number of revision and evaluation exams. The student performance in the unit evaluation exams are registered. each unit has a weighting factor which is used in computing the student grading point average. The presenter provides a user friendly interface to present the courseware in a predefined order but it allows order readjustment and facilitates selfpaced learning and the exam answering. The courseware editor, through a very simple command set, can help the instructor to list the course students, edit course units and tests and keep track of the test grades of each course student.

INTRODUCTION

Computer Assisted Education (CAE) is defined as the use of computer to assist in all or some branches of education such as administration or learning process. Computers may be taught as a science by itself and may also be used as a tool in the administration process, since student registration system, student and staff attendance system, and other affairs can be computerized. Computers can also assist the instruction process as courseware preparation and presentation. Computer Assisted Instruction (CAI) package may be identified as an interactive learning tool for authoring and learning so the student may interact with the presented material based upon individual needs by choosing his or her own sequence and direction through the material. The educational problems are the main motive for CAI systems creation as increasing the traditional method operating cost and waste time.Other existent problems are increasing the number of learners with variant abilities and decreasing the well trained instructors especially in a certain field. So, CAI can be processes. There are some problems with CAI especially in the authoring system.

Considered as a solution to improve the quality of instruction at lower cost. CAI can offer the interactivity with the student, different learning methods,and more accurate evaluation.

These problems create the need of Intelligent Computer Assisted Instruction system (ICAI). ICAI is based on expert system approach and its contents and instructional strategies take the form of semantically related concepts and inference rules. The characteristics that distinguish ICAI systems from CAI systems are mixed initiative dialogue, semantic or knowledge nitwork, and error diagnosis. So, ICAI systems can provide a much finer degree of individual instruction

and allow the learner a greater control over the instructional interaction than with conventional CAI. Some courseware preparation language exists as BBC, BASIC, LOGO, and PILOT.

This paper presents intelligent courseware system (ICS) as an intelligent computer assisted system. Section 2 introduces the model and an intelligent courseware editor (ICE) design and implementation is discussed in section 3.

2. I C S MODEL

ICS consists of four modules: Registrar, grader, courseware editor, and presenter as shown in Fig.1. The registrar keeps track of the students and their performance in the course exams. The course exams are graded by the grader module which updates the student database set up by the registrar module. The presenter is responsible for material presentation according to predetermined order. The courseware editor facilitates the material preparation. The system users are the instructor (the courseware author) and the student. ICS model satisfies both user types requirements.

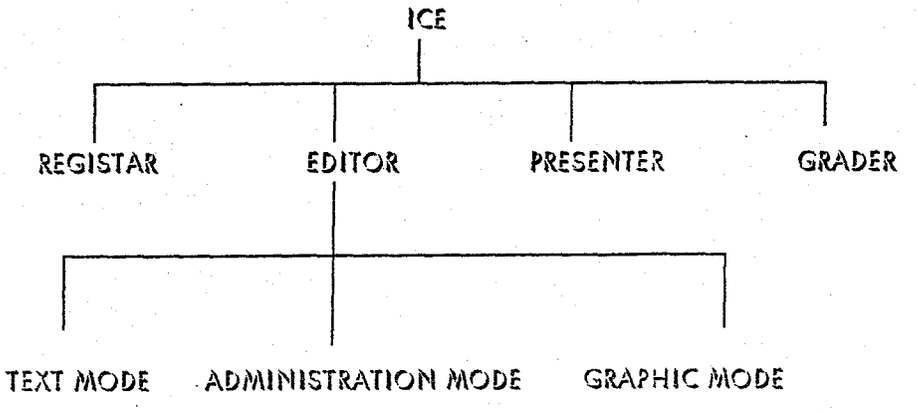

Fig.(1) : The ICS Organization

2.1. The Courseware Submodel

The courseware submodel supports the following concepts:

- The courseware may be divided into chapters, the chapter into sections, the section into subsections, the subsection into subsubsections, and so on.
- The lowest subdivision level object is called unit, it is a nondecomposable object and it is the terminal of the course hierarchy (Fig. 2).
- The units only have course material and the chapters and the subsections do not have material but abstracts.
- The chapter, section, or unit body always start by its id in a separate line preceded by colon (for example:1.2.3 is the beginning line of unit 3 of section 2 of chapter 1).
- There are two types of exams: revision exams and evaluation exams. Only the evaluation exams are used to update the student grading point average (gpa).

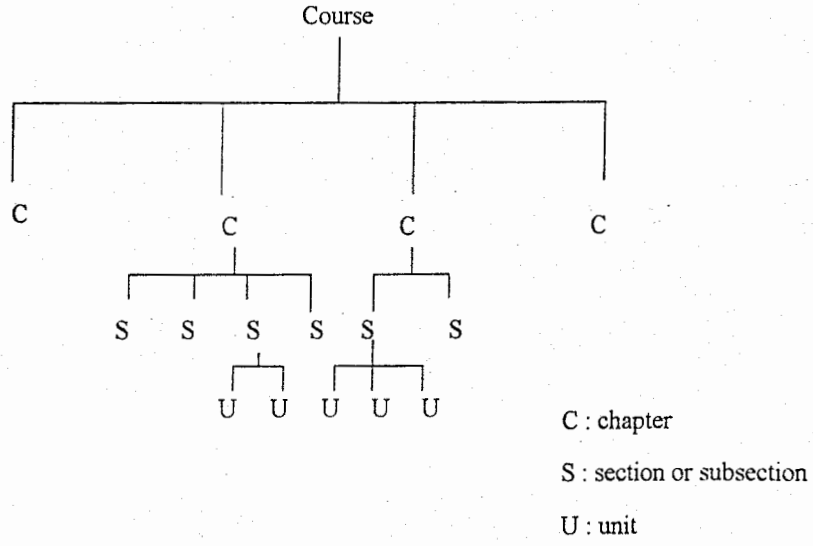

Fig.(2) : Course Material Hierarchy

- The unit may be followd by any number of revision and valuation exams.
- Each object (chapter, unit, section, and subsection) has a record (or line) in the unit file with unique id, title, abstract, type, classification, and complexity level as shown in table 1.
- The unit type may be coded as chapter, section, unit, revision exam, and evaluation exam.
- The unit classification may be coded as mandatory or optional.
- The unit complexity level may be coded as elementary, intermediate complex.
- Up to six keywords are stored with each unit and they are used to search for units.
- The unit is the smallest unit to save, load, and display.
- Each unit has a weighting factor used in computing the student grading point average (GPA).

Table (1): Unit identity data

Atribute	
uid	unit unique is
utitle	unit title
useq	unit version sequence number
uweight	unit weighting factor
uabstrat	unit abstract
utype	unit type
uclassification	unit classification
ucomplexity	unit complexity level
ulimit	maximum time allowed (only for evaluation exam)
uword	up to six keywords
ufile	file name in which the unit is saved
udisk	disk id in the the unit is saved

Table 2: Course hierarchy (recursive table)

Attribute	meaning
uid	unit id
pid	parental unit id

- A unit may have more than one version where all versions have the same unit id but different sequence numbers. The version with the lowest sequence number is the default one to display. Another version may be displayed instead according to the studentperformance. All unit versions contain the same material but in different presentations (e. g. more examples and/or graphs) and they may have different leels of complexity. A unit may have prerequisites, as other units have to be visited at least once before the student can study this unit. A unit and its prerequisites are expressed in Table 3.
- A user friendly courseware editor should be available to the instructor for all authoring purposes.
- Order of unit entrance may be independent of order of unit presentation.

Table (3): Unit Prerequisites (recursive table)

attribute	meaning
uid	unit id
rid	prerequisite unit id

2.2. The User Submodel

- There are two types of users: the instructor and the student. The instructor acts as the system administrator having the capability to edit and display any data in the system.The student is a regular user having only the right to display the course material and take exams.
- Each user has a user name (login name) and password to login in the system. A special login name, instructor, is reserved for the system administrator. The instructor is supposed to be the only one who knows the instructor login password.
- Each user has a record (separate line) in login file as shown in Table 4.
- The instructor can register a student by editing the login file and inserting the user line.
- Each student has a unique id, name, login password, average grading point average.
- The student cannot change the unit presentation order, the grading policy, the course material, and the student grades.
- A student may check only his or her grades not other students, grades. The unit or revision exam may be studied any number of times.
- A student may have the right to exit from any point of the unit and the number of unit revision is not incremented unless the student exit from the last point of the unit.
- A unit and revision exams may be presented any number of times, but an evaluation exam is presented to a certain student only once for evaluation and then any number of times for just revision. The evaluation exam is presented at the first time for a

certain amount of time set by the instructor, but the student can exit the exam any time.

- The student can ask for any unit or any revision exam any number of times and they are displayed from the beginning.

Table (4): User Login Table

attribue	meaning
sid	student unique id
sname	student name
password	encrypted student password
class	student class
gpa	grading point average (computed attributed)

2.3. Grader

The grader is an attached subsystem responsible of computing the student grading point average according to the following concepts. Table 7 shows student unit interaction as it follows.

- For each registered student and for each reviewed unit by the student, the number of unit revisions and the unit evaluation exam student performance are reserved.
- The student is allowd to have a unit evaluation exam once for evaluation and in terms of evaluation exams, a version is considered as if it were a distinct unit.
- The instructor may be able to change the grading policy to recompute the student gpa by changing the weighting factor of unit exams.

- Each student has a score for each unit he or she passed its evaluation exam. If the student passed more than one evaluation exam of different versions of the same unit,the unit score is computed accroding to the formula that follows.
- For each student, grading point average is computed for any logical combination of units: unit (there may be more than one version for the unit), subsection, section, chapter. The final GPA can be computed for all already done units. GPA is computed according to the following formula:

$$gpa = sum (x(i)\ w(i)\ e(i)) \text{ for all } i$$

where x (i) = 1 if unit i exam is done by the student,
= 0 otherwise.
w (i) = unit i weighting factor, and
e (i) = unit i exam score

3. THE PRESENTATION MODULE

The student mainly interacts with the presentation module (PM). The user must authenticate himself or herself to PM once it is invoked by entering the login name and the password. The student is initially given a password by the instructor but it can be changed later. The opening screen is shown in Fig. 3 where information about the already visited units are displayed, then the student is asked to enter a unit to display. If ENTER key is pressed, instead of any number, PM starts to display a unit according to a certain order. The student may press ESC to back to PK main menu or q to quit. The PM main menu has five options: list, presentation, exam, gpa, and exit as shown in

Fig. 4. The list option lists a unit, all done units, or all chapter (or section) units. For example, if the student selects chapter units option of submenu of list and then enters 1.4, PM responds by listing all descendent subsections and units under section 4 of chapter 1. The listed information about each subsection or unit are unit id, version number (in case of unit), unit title, number of unit revisions, and gpa. The units may be searched by keywords.

Welcome to ICS Tutor

user name :
password :

You already have revised the following units with GPA=2.9 :

unit-id	version	unit-title	number-of-revisions	score (of 100)
1.1	1	relational model	3	70
1.1	3	relational model	1	90

unit number:

(enter unit number, enter for automated unit select, esc to back to ain menu, q to quit)

Fig.(3): The Presenter Opening Screen

The presentation option either presents units in a certain order or allows the student to enter a unit id to be presented. If a section id is entered (any non unit object), its abstract is displayed. If a unit number is entered and any of this unit prerquisites have not been visited, the student will be informed by all unvisited prerequisites and the unit will not be displayed. If a unit number is entered, the file name which contains the unit is known from the unit identity table, the file is located under the courseware directory, and then it is searched for the unit (it may contain more than one unit). PK assumes that the course matterial files are loctated under courseware directory immediately under the current directory, but the student can handle this restriction by assigning the full path name of the courseware directory to courseware shell variable in the PM profile file (named .pm). For example, courseware = /usr/app/course/courseware is given in unix shell format.

The presentation order may be automated dependent on the prerequisites of each unit and which unit version is presented is dependent on the student performance. This dependency are expressed in production rules.

Rule Example.

```
/* The lowest sequence number version is presented by default.*/
If the student performance in this default version is poor
        (less than certain level)
   Then the version with next sequence number is presented.
End if
/* next rule is dependent on the previous rule */
If the student is doing a little bit better (higher score in
   this version ) and this unit is prerequisite for more
```

than one unit,

Then the version with next squence number is presented

End if

The exam option allows the student to take an exam. An evaluation exam is taken in a certain amount time for the first time but the student can quit before this set time. Only the first time evaulation exam taking is considered for evaluation and gpa computation. Any other time taking is just for revision like any revision exam and the time constraint is released. After the student is done with the exam, it is graded and the student is informed with the score and the incorrect answered question (if any) with the correct answers.

1. list
   1.1 all
   1.2 done units
   1.3 chapter (section) hierarchy
   1.4 keywords
2. presentation
   2.1 automatic
   2.2 manual
   2.2.1 unit id
3. exam
   3.1 revision
   3.2 evaluation
4. gpa
   4.1 unit
   4.2 section (chapter)
   4.3 all done
5. exit

Fig.(4): The Presentation Module Menu

4. THE COURSEWARE EDITOR

The courseware text and graphics editor (ICE)has three modes text, admininstration,and graphics modes as shown in Fig.1

Table (4): User Login Table

attribue	meaning
sid	student unique id
sname	student name
password	encrypted student password
class	student class
gpa	grading point average (computed attributed)

Table (5): Student Unit Interaction

attribute	meaning
sid	student id
uid	unit id
useq	version sequence number
uvn	number of unit revisions
ues	unit evalution exams score
uet	the time spent during the exam

4.1 The Text Editor

The course material may be divided into any number of chapters where each chapter has a unique number and also the chapter itself may be subdivided into sections where each section has a unique

number within the chapter, and so on. The lowest level subsection is called unit. The unit is nondecomposable and the smallest object to load, save, and display. The course material hierarchy is shown in Fig. 2. The numbering determines the order in which the objects (chapter, section, subsection, or unit). are displayed. The object with the lowest number is displayed first. The sections of a chapter are also displayed in order of their numbers within the chapter. The whole course may be considered as one chapter if it is not divided and the chapter may be considered as one section if it is not divided. The text editor has two modes of operations: administration and update modes. In administration mode the instructor types only commands for specific actions with specific syntax. Each command must be preceded by a colon. In update mode the instructor can edit a text.

4.2. Administration Mode

ICE is specialized editor in courseware preparation and it offers command to support this feature. The administration commands are given in Table 6. A course chapter, a section, or a ubsection(unit) may be marked as a current course material part (object). Some of the editor commands are applied to the current object. The cnum argument of c command is the chapter number to be the current chapter to deal with. If a subsection is the current,s command without argument is submitted to mark the upper level section or subsection (the parent) as the current. It can be repeated until the most upper level (the chapter) is reached and the same result may be got directly by issue c command without arguments. you may move from one chapter to another by c command. The section number as an argument can be expressed as an unqualified (relative addressing) or qualified (absolute addressing)Unqualified section number is just an integer to

represent a subsection of the current object (chapter or section) and qualified section number is a series of integers separated by dot. For example, (s2)command marks subsection 2 of the current object (chapter or section)as the current subsection and (s1.2.3)refers directly to subsection 3 of section 2 of chapter 1 and marks it as the current object. The course chapters are stored in files. A file may contain one or more than one chapter. A nondecomposable objects have to be saved as a whole, but subsections of a section or sections of a chapter may spread over more than one file. In other words, a file should start with the beginning of chapter, section, subsection, or unit. The load command allows the user to load a chapter or section from a given certain file. The section number as an argument to the load command has to be qualified. If a chapter number is given (unqualified number), then the chapter and all its sections saved in the given file will be loaded and also the section and all its subsections stored in the file are loaded. The user can drop a chapter, a section, or a subsection. All changes are done in the memory and has to be saved in a file. There is a command to check certain file contents if it is in the course format, display the chapters and sections numbers which are stored in the checked file (without loading). During this context, square brackets surround optionals and pipe symbol means either side, so c and s commands may be expressed as :

```
: u  [ cnum   | # ]
: s  [ snum   | # ]
: r  [ cnum   ]
: d    cnum   | snum
: l    cnum   | snum          file _ name
: f  file _ name
: t  file _ name
```

snum : = snum | snum.snum
num : = cnum | snum

The system gives an error message if a user issues s with no current chapter, if cnum or snum are not whole integer or if they are not in the proper range. A chapter number and a section number within a chapter must be unique, otherwise an error message results.

Table 6: Administration Mode Commands

command arguments	**Function**
: c	back to the chapter as the current object
cnum	select course chapter cnum to be the current
#	display the course chapters order
:s	one upper level subsection to be the current
snum	select section number snum to be the current
#	display the sections order within a chapterq
: r	reorder (renumber) the chapter
cnum	reorder the sections of the specified chapter
:d num	drop chapter or section
:l	filename num load chapter or section from the given file
:a file name	save the memory copy to the given file
: f file name	test the file format
: g	graphics mode

4. 3. Update Mode

ICE has a flexible screen editing commands allowing the instructor to edit chapter or section text. These command are showen in Table 7.

Table 7: Text Screen Editing Commands

command	function
PgUp	page up (10 lines up from the current line)
PgDn	page down (10 lines down from the current line)
End	move cursor to the last char of the current line
Home	move cursor to the first char of the current line
Insert	insert just left to the current char
Del	current char deletion
Ctrl-PgUp	move the cursor to the current chapter beginning
Ctrl-PgDn	move the cursor to the current chapter end
Ctrl-B	move the cursor to the current block beginning
Ctrl-E	move the cursor to the current block end
Ctrl-M	mark a block of text
Ctrl-V	move the marked block to the current position
Ctrl-C	copy the marked block to the current position
Ctrl-D	delete the marked block
Ctrl-S	search for a specific text
Ctrl-R	search and replace text

6. CONCLUSION

An intelligent courseware system was presented. The system model is composed of the courseware submodel and the user submodel and other modules like the courseware text and graphics editor, the presententation module. The model satisfies the instructor and student requirements. The instructor as a system administrator can create student accounts, edit courseware and student grades. The system has an authentication subsystem allowing a secure login to the system. The

courseware is divided into units forming a hierarchy and the student can consult a unit any number of times. Each unit is followed by arbitrary number of revision and evaluation exams. The student performance in the unit evaluation exams are registered. Each unit has a weighting factor which is used in computing the student grading point average. The presentation module provides a user friendly interface to present the courseware in an order dependent on student performance and unit prerequisites. The courseware editor, through a very simple command set, can help the instructor to list the course students, edit unit and exam material and keep track of the test grades of each registered student.

REFERENCES

1- C. J . Date, "An Introduction To database Systems," Volume I,Fifth Edition, Addison-Wesley,1990.

2- B. Gavish and H. Pirkul, "Computer and Database Location in Distributed Computer Systems," IEEE Trans. on Computers, Volume C-35, Number 7, July 1986.

3- Kak, "Data Security in Computer Networks," Computer, Volume 16, Number 2, Feb. 1983.

4- A. Barr,E.Afeigenbanm,"The Handbook of Artificial Intelligience," Vol. 1, William Kaufmann Inc.,1981.

5- R. Forsyrh, Ed., "Expert Systems, Principles and Case Studies," 2nd ed., Chapman and Hall Computing, 1989.

6- P. A. Subrahmanyam, "The Software Engineering of Expert Systems: is Prolog Appropriate?," IEEE Trans.on

Software Engineering, Vol. SE-11, no. 11, Nov.1985,pp. 1391-1400.I. M.

7- Begg,"An Intelligent Authoring System," IEEE Trans.on Software Engineering 1985.

8- M. David Merril,"An Expert System For Instructional Design"IEEE Trans. on Software Engineering, Vol. 2, No. 2,1987.

9- T.Miccerit and others,"Must Computer Courseware Evaluation Be Totally Subjective," British Journal Of Education Technology, Vol. 20, No. 2, 1989.