Knowledge Maps for Intelligent Questioning Systems in Engineering Education

**James F. Leathrum, Jr., Oscar R. González, Stephen A. Zahorian,
Vishnu K. Lakdawala**

**Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529**

Abstract

The development of a hierarchical knowledge map to be used with an intelligent questioning system is described in detail in this paper. The purpose of the intelligent questioning system is to improve the educational process in engineering courses by allowing students to learn more in less time, to understand more deeply, and to enjoy their learning experience. Key elements of this system are a question model and an adaptive question management system that uses a hierarchical knowledge map to direct the learning process based on the student's degree of understanding of individual or grouped concepts. Although there are several online computer-based questioning systems, they typically have no built-in help, no guidance if questions are answered incorrectly, no method for selecting questions based on the students needs, and no comprehensive monitoring of a student's progress through a knowledge map of the course and the overall curriculum.

The knowledge map is a formal representation of the knowledge to be imparted to students in a program of study. In addition, the knowledge map has sufficient structure to capture a model of each student's progression. It provides a graphical map of the concepts that a student has learned and the degree of understanding for each concept. At the highest level, the knowledge map represents the architecture of the entire curriculum. In the second level, the knowledge map represents each course as an interconnection of modules. The third level gives the architecture for the module's concepts and their relations. The structure representing each concept will specify the types of questions that are relevant. An adaptive guidance system will then be able to select a particular question from the question database depending on the student's current level of understanding. Grouped concepts will allow a comprehensive assessment at each level.

1. Introduction

Over the past several years, considerable effort has been devoted to research in the area of technology-enhanced education. Progress has been made, addressing a variety of educational needs, ranging from supplements to existing "traditional" courses, to complete on-line courses, to complete on-line programs. Despite all this effort, hype, and even product development, most of the courseware material available for use at the college level is still not judged to be as effective as a professor lecturing and leading discussions with students. Most of the work is driven more by convenience factors in continuing education due to the increasingly mobile student population ("anytime, anyplace, anywhere," education), perceived economies of scale (some educational administrators envision that a few "super star" faculty could each teach thousands of students), and to a certain extent just because the technology is available. Nevertheless, faculty working on a daily basis with students continue to pursue the dream of using technology to improve the educational process, allowing students to learn more in less time, to understand more deeply, and to enjoy their learning experience more.

The present state of affairs in cyber education leads us to the conclusion that focused yet long-term research is needed to move forward. The particular focus of our research is to develop a methodology for an intelligent computer-based questioning system to be used in engineering programs. One of the unique aspects of the engineering field is the need for questions of many types--- basic information, analytical problem solving, design techniques, and finally open-ended design problems. These different types of problems will be addressed in our methodology.

The primary goal of this paper is to present the knowledge model that will direct the learning process based on the level of understanding of the student in each topic area. The key components are a formal representation of the course topics, an architecture to link the topic's representation with a question model and a set of rules for intelligent questioning and assessment. In particular, this paper's focus is to describe the architecture of the databases needed to support the intelligent questioning system under development.

2. Background

The potential for computer-based questioning systems has already been recognized and commercial products are already available (see, for example, WebAssign[6] and WebCT[7]). Such systems are well developed for courses with large audiences such as freshman physics. Primary benefits include immediate feedback given on every question attempted; the options to easily vary parameter values for each question, and automated record-keeping options. Typically higher-achieving students prefer these systems for homework questions to more conventional methods. However, most current systems are little more than a computerized version of homework problems found in a typical textbook. The only feedback given to the student is "correct" or "incorrect" for each question answered. There is no built-in help, no guidance if questions are answered incorrectly, no individual student-centered method for selecting questions based on a students needs and background, no accompanying simulations to illustrate key concepts, and no comprehensive monitoring of a student's progress through a concept map of the material. The intelligent questioning system under development will address these deficiencies by adding to the "intelligence" underlying automated questioning systems. The goal is to determine how to build a questioning system, which will appeal to and benefit a wide range of learners with a large range of capabilities.

A very good intelligent on-line physics homework system is Andes[5]. This work-in-progress is being developed by personnel of the Physics Department of the United States Naval Academy and of Computer Science from the University of Pittsburgh. In Andes, the students can solve the problems directly in the web browser. The solution steps consist of defining the variables, drawing free body diagrams, writing and solving the equations. If the instructor allows it, Andes will inform the students of problems with a free body diagram and equations, and even solve the set of equations. Andes can find the mistakes because its knowledge base consists of a set of rules to solve the physics problems. Andes is smart enough to determine multiple solution paths, if they exist. One of Andes' strengths is the tutor system. Depending on the instructor's options and the student's past experience the tutor can explain the mistakes and can assist with the solution. Andes' also has a very good student assessor system. By observing the students use of rules, the assessor forms a modified Bayesian net, computes probabilities of the student's knowledge, proper use of a rule, and expected solution[3]. This information is used by the assessor to tell the tutor system how to help the student. Two intelligent tutor systems for electrical circuits were reported by Ahmed and Bayoumi[1] and Yoshikawa, et. al.[8]. These tutors do not develop a student model and can only provide limited feedback to the students.

3. Intelligent Questioning System Overview

The Intelligent Questioning System is comprised of three basic components: the User, the Knowledge System and the Question System as shown in Figure 1. An appropriate user model drives the system. The Knowledge System presents the user with his current status in the system by use of a graphical map. This gives the user a set of options of concept areas that he or she may work on. The selection of a concept area defines a knowledge key that is presented to the Questioning System. The Questioning System then identifies an appropriate set of questions within the concept area. A question is then randomly selected and presented to the user with random parameters. On answering the question, the user is provided immediate feedback. Should the Questioning System identify that the user needs some help, it initiates this process. Help may come in several forms:

- links to material for review,
- suggestions, and
- breaking down the problem into smaller steps.

When the user completes or chooses to leave the concept area, the Questioning System provides an assessment measure back to the Knowledge System. The Knowledge System then updates the user's status in the system, potentially identifying a new set of concepts that the user may work on.

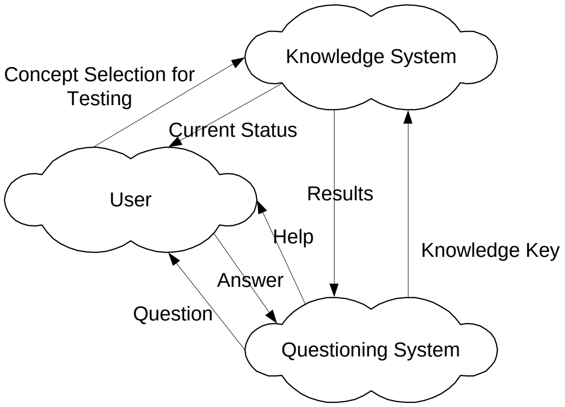

Figure 1. Intelligent Questioning System Concept.

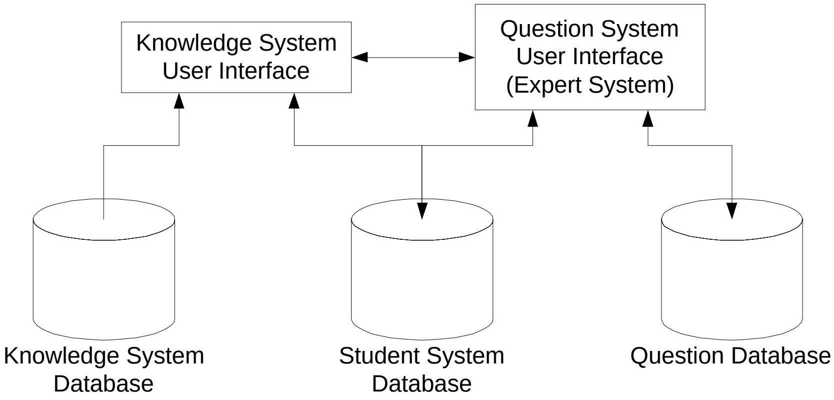

Figure 2. Intelligent Questioning System Structure.

To support this concept, several databases are required as demonstrated in Figure 2. The question database contains the complete set of questions. The student system database maintains information about which questions (and associated parameters) the student has attempted and the assessment measure for each question. The knowledge system database maintains the structure of knowledge flow in the form of knowledge maps. The knowledge system database is the primary topic of this paper.

4. Knowledge System

The knowledge system is a hierarchical collection of knowledge maps creating a tree. The map structure is maintained in a database that is referenced to direct the flow of learning. The knowledge maps are directed graphs guiding the assessment process by assuring mastery of prerequisite topics before advancing. The leaves of the tree are nodes acting as portals to the questioning system used to assess the student's grasp of the knowledge. Other vertices in the tree define the next hierarchy map in the knowledge system.

Nodes are points in the knowledge system where action must be taken to advance. The action may be to move to a sub-map, or to interact with the questioning system to assess understanding. A node in the system may be one of three types:

- *Map node*: defines a new knowledge map (denoted by a rectangle in our diagrams)
- *Question node*: defines the interface to the questioning system (denoted by an oval in our diagrams)
- *Null node*: an empty node used as a placeholder in a map (denoted by a triangle in our diagrams)

An example structure for a knowledge system is demonstrated in Figure 3. Note that the system starts with a single map node at the root of the tree. That node defines a map, which in itself may define other maps. Questioning content is defined by the locations of the question nodes.

Each node has an entry in the database maintaining information about the node and its relationship with its knowledge map and with the whole system. All nodes contain the following fields:

1. *List of predecessor nodes* – nodes that immediately precede the node.
2. *List of successor nodes* - nodes that immediately succeed the node.

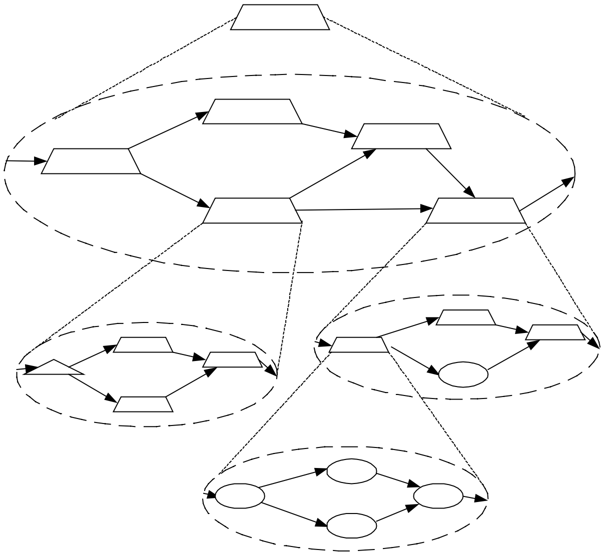

Figure 3. Knowledge System Constructed from Hierarchical Knowledge Maps.

3. *Parent map* – the map to which the node is a member.
4. *Weight* – the relative weight of the node compared to other nodes in its map for computing an assessment measure for the parent map. For null nodes, the weight must be zero.
5. *Minimum score* – the minimum score required to satisfy completion of the node. For null nodes, the minimum score is zero.

The basic knowledge system is generally static in nature, defining the flow of knowledge that the student must gain. The structure may change as core knowledge in a field that changes frequently such as computer engineering dictates this ability. But these changes should not occur more frequently than on a semester-by-semester basis.

The Student System Database maintains some information about individual students status in the system. This database is very dynamic in nature as it is updated each time a student answers questions. The database contains the following information on each student for each node in the system:

1. *Score* – The accumulated score for the node. It is defined by:
   - Map node:

$$score = \frac{\sum_i score(n_i) \times weight(n_i)}{\sum_i weight(n_i)} \quad where\ n_i\ denotes\ node\ i\ in\ the\ map$$

   - Question node: defined by the question system.
   - Null node: zero.
2. *State : {completed, not reached, in progress}* – Each node in the system can be in one of three states:
   - *Completed* – The student has completed the node defined by:

$$score > minimum\ score$$

- *Not reached* – The student has not started the node.
- *In progress* – The student is in progress of completing the node.
- *Revisit required* – A later question node has identified this node as an area of concern. The student must revisit this node and demonstrate a new mastery of the topic.

Other information is maintained about the student's status in terms of which questions he or she has attempted out of the Question Database. This is beyond the scope of this paper and will be discussed in a future paper.

Each of the three base node types is discussed in more detail.

4.1. Map Node

A single map node is defined by a set of nodes and the precedence relationship between the nodes defining a directed flow graph. A map node is of type node, thus inherits all of the properties of a node. Each map node also contains:

1. *Inherited properties from Node*
2. *List of nodes* – the set of nodes which constitute the map
3. *Entry node* – The first node visited in the map (there may only be one entry node). The entry node must be a member of the list of nodes. In the case where there are several topics in the map, which may be started concurrently, the entry node is defined as a null node which is automatically satisfied allowing the successor nodes to be attempted immediately. This is the primary use of null nodes.
4. *Exit node* – The last node visited in the map (there may only be one entry node). The exit node must be a member of the list of nodes. It is appropriate to make this node a question node, which is a cumulative assessment of the total comprehension of knowledge in the map. This node may also be a null node in the case where there are multiple nodes in the map, which are unrelated on completion. This concept primarily supports the curriculum map discussed later. Completion of the exit node triggers the computation of the score for the map node and thus possibly may change its state.

4.2. Question Node

The question node provides an interface to the questioning system. Each question node contains:

1. *Inherited properties from Node*
2. *Knowledge key* - a key defining the knowledge area to be tested by the questioning system.

4.3. Null Node

The null node is simply a placeholder in the structure to support multiple entry and exit nodes.

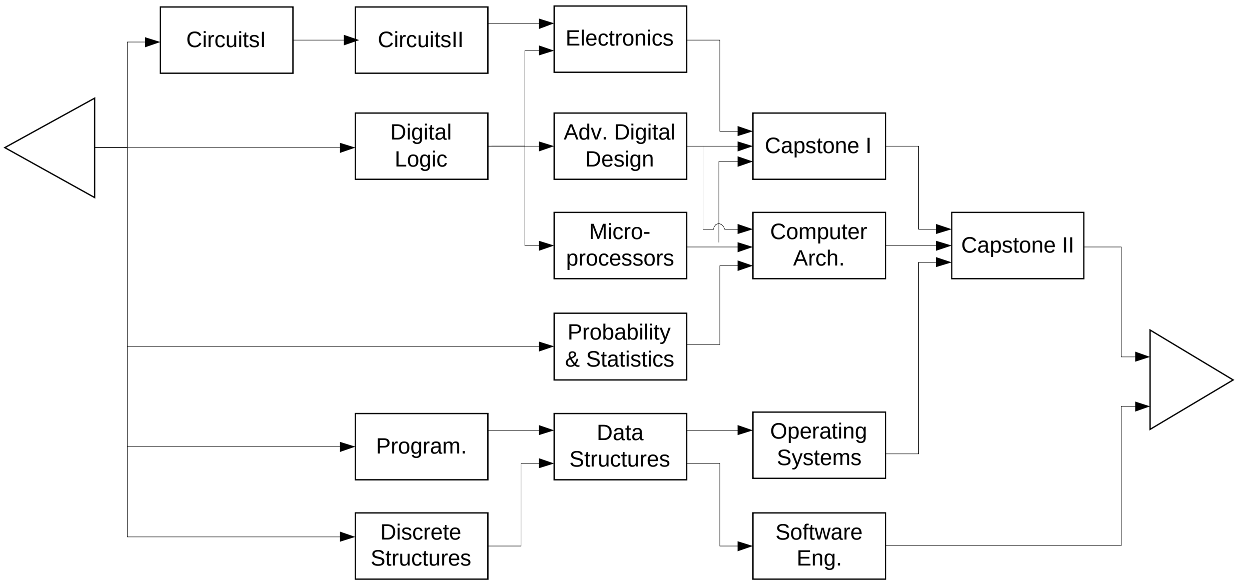

Figure 4. Knowledge Map Representation of the Computer Engineering Curriculum.

5. Example – Knowledge System for Computer Engineering

The Computer Engineering program at Old Dominion University is currently under development as a web enhanced curriculum. This serves several purposes including, improving the current delivery of material, improve retention of information, and support distance learning efforts such as a new program recently agreed to with Rajagiri College in Cochin, India. [Zahorian, et. al.] As such, the curriculum is being reviewed to identify how to best capture the

structure of the curriculum electronically. To this end, the current form of the curriculum is presented here as an example of the knowledge system. The system is viewed from the top down, first looking at the curriculum, then the course level, and then topics within the course.

5.1. Curriculum Example

At the top level, the curriculum itself is captured as a knowledge map. This map for the Computer Engineering curriculum at Old Dominion University is presented in Figure 4. Note that a single entry point is defined despite the fact that there are several entry points to the curriculum itself (i.e. there is no prerequisite structure between Digital Design and Circuits).

5.2. Digital Logic

Each course in the curriculum can then be broken down into a knowledge map defining the flow of information within that topic. Figure 5 demonstrates how the Digital Logic course can be represented as a knowledge map. Each node identifies a topic area that is in itself defined by a knowledge map. Advancement in the map requires satisfying the requirements of each topic area. The exception is the final node, *Design Problems*. This node is actually a question node. The purpose of this node is to give a single assessment point to measure the integrated understanding of the material. This node is analogous to a final exam in the course. This node gives direct feedback indicating that some areas require revisiting. It should be noted that the general structure of the knowledge map in Figure 5 is loosely based on the textbook by Nelson, et. al.[4] currently used at Old Dominion University. However, it is dangerous to tie the knowledge system too closely to a textbook as the system can become outdated by a change in the preferred textbook.

5.3. Sequential Circuit Synthesis and Analysis

One topic in Digital Logic is Sequential Circuit Synthesis and Analysis. This topic is represented by the knowledge map in Figure 6. Each node provides an interface to the question system such that the user must pass a set of questions before being allowed to advance.

5.4. Student State Information Example

An example is developed to demonstrate the type of information maintained for each student to define his state in the system. Consider a student currently eligible to work on nodes Circuits II, Digital Logic, Probability and Statistics, and Data Structures in the curriculum shown in Figure 4. The student may have started work on Digital Logic and Data Structures. If this were the case, then the state information for the nodes would be that given in Table 1. Any node that is marked as Not Reached would have no further information retained for the child knowledge maps. However, the nodes marked as completed or in progress would maintain similar information for each child knowledge map. The end result of the state information is to identify which question nodes are currently available to the student as indicated by a state of *In Progress* or *Revisit Required*.

Table 1. Student State Information for the Curriculum Knowledge Map in Figure 4.

Node	**State**
Circuits I	Completed
Circuits II	In progress
Electronics	Not reached
Digital Logic	In progress
Advanced Digital Logic	Not reached
Microprocessors	Not reached
Computer Architecture	Not reached
Probability and Statistics	In progress
Programming	Completed
Discrete Structures	Completed
Data Structures	In progress
Software Engineering	Not reached
Operating Systems	Not reached
Capstone I	Not reached
Capstone II	Not reached
Entry Node	Completed
Exit Not	Not reached

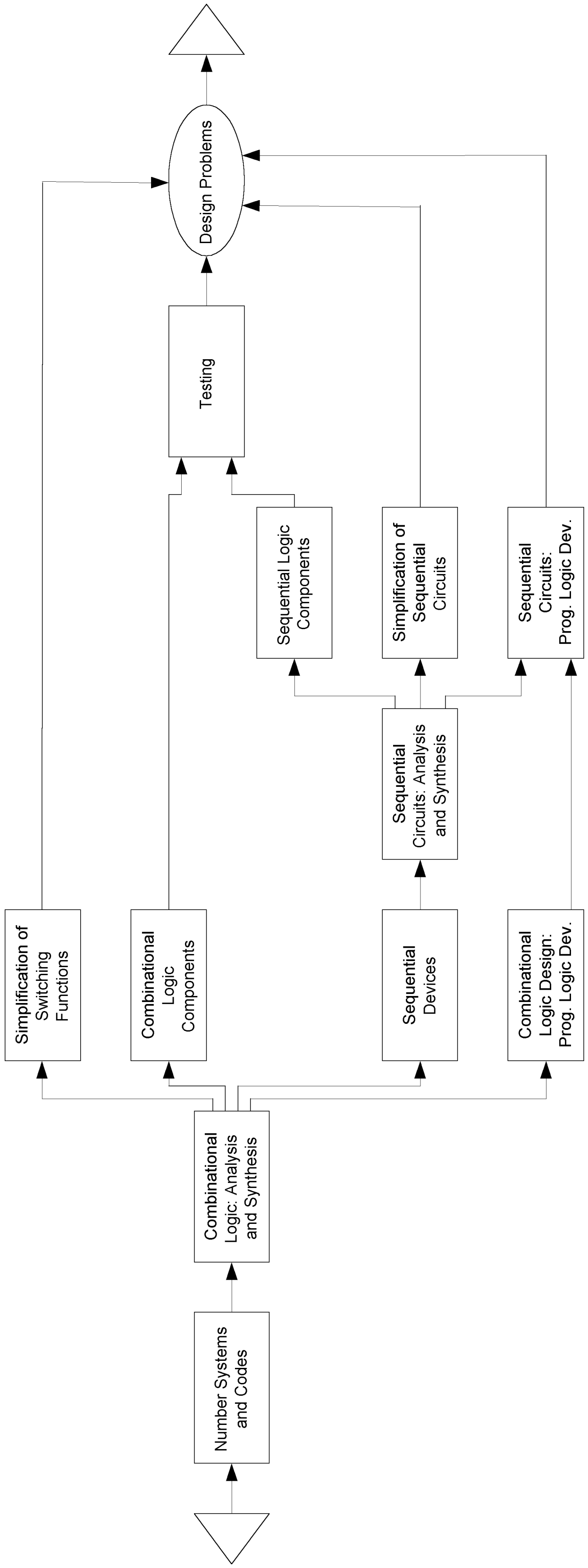
Number Systems and Codes
Combinational Logic: Analysis and Synthesis
Simplification of Switching Functions
Combinational Logic Components
Sequential Devices
Combinational Logic Design: Prog. Logic Dev.
Sequential Circuits: Analysis and Synthesis
Sequential Logic Components
Simplification of Sequential Circuits
Sequential Circuits: Prog. Logic Dev.
Testing
Design Problems

Figure 5. Knowledge Map Representation of a Course in Digital Logic.

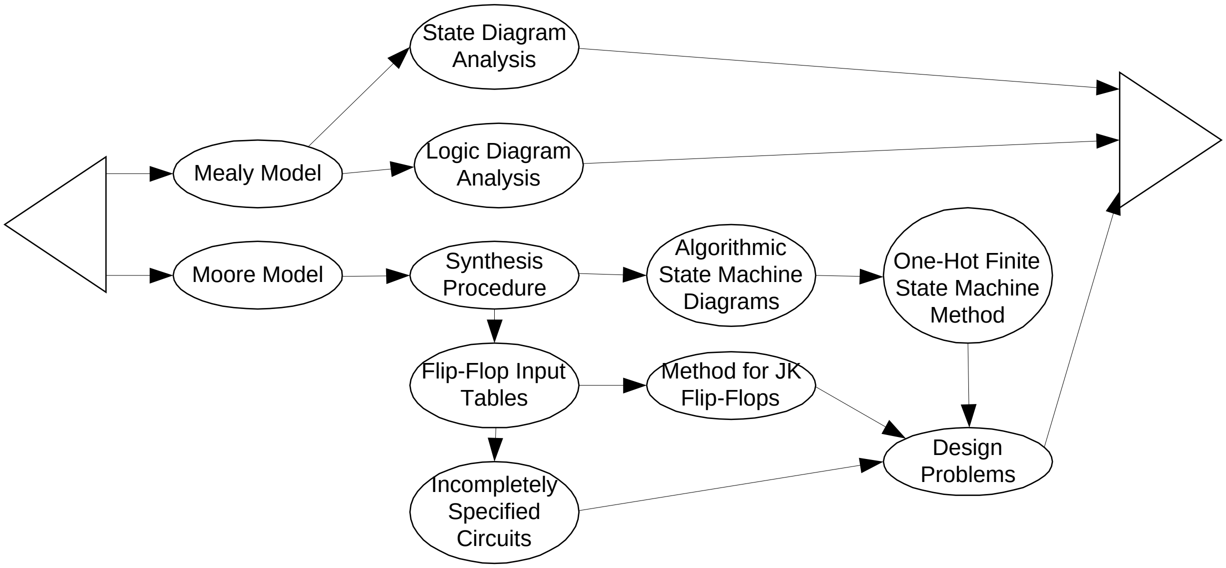

Figure 6. Knowledge Map Representation of Sequential Circuit Synthesis and Analysis.

5. Future Work

This paper presents one piece of a work in progress. At this point in time, it is the most complete component of the project, though it is realized that the future development of the other components will require updates to this structure.

Also under current development is an initial question database. The set of beta questions are simple in nature. Future development will support a variety of question types and random parameters for reuse of questions. The questions will also be mapped to the knowledge map by use of the knowledge key.

The biggest component remaining is the questioning system. This system will be an expert system that will intelligently select questions based on past results and will attempt to identify areas of concern where the student demonstrates weaknesses. In the case of identified areas of concern, several options may occur. First, the system will attempt to zero in on specific concerns by asking more questions. Second, the system will suggest review material, either written material or electronic material maintained in modules supporting technology enhanced learning[9,10]. Lastly, the system will mark nodes in the knowledge system as *revisit required* so that the student will have to redemonstrate a proficiency in the subject matter.

6. Conclusions

The success of a good web based curriculum is a good assessment process. This paper presents the data structures necessary to capture the knowledge flow required in an engineering curriculum. This structure will allow the development of the Intelligent Questioning System as discussed. The knowledge system is a formal representation of the knowledge flow in a curriculum. This provides the necessary framework to allow a questioning system to select appropriate questions. Also, if the questioning system detects a problem in the student's learning, asking multiple questions covering multiple knowledge keys, allows the questioning system to isolate areas of concern.

The goal of this work is to support a web based Computer Engineering curriculum, both domestic and international, as well as to improve the learning process for the classical student. The system will give immediate feedback and assistance to the student, an improvement over the delays required with classical assessment measures such as homework. Thus, while still a work in progress, it is believed that the success of this project will provide an improved learning environment and the capability to support distance learning.

Bibliography

1. Ahmed, M. & Bayoumi, M., An artificial intelligent instructor for electrical circuits, Pro-ceedings of the 37th Midwest Symposium on Circuits and Systems, pp. 1362-1365, 1994.
2. Zahorian, S., Albin, S., & Swart, W., Global Engineering Education: A partnership between Rajagiri College (Cochin, India), and Old Dominion University (Norfolk, VA), submitted for presentations at ASEE 2001.
3. Martin, J. & VanLehn, K., Student assessment using Bayesian nets, International Journal of Human-Computer Studies, vol. 42, 1995.
4. Nelson, V., Nagle, H., Carroll, B., & Irwin, J., Digital Logic Circuit Analysis and Design, Prentice Hall, Englewood Cliffs, NJ, 1995.
5. Schulze, K., Shelby, R., Treacy, D., & Wintersgill, M., Andes: A coached learning environment for classical Newtonian physics, Proceedings of the 11th International Conference on College Teaching and Learning, Jacksonville, Florida, 2000.
6. URL: http://www.webassign.com; WebAssign.
7. URL: http://www.webct.com; WebCT.
8. Yoshikawa, A., Shintani, M., & Ohba, Y., Intelligent tutoring system for electric circuit exercising, IEEE Transactions on Education, vol. 35, pp. 222-225, 1992.
9. Zahorian, S., Swart, W., Lakdawala, V., Leathrum, J., & González, O., A Modular Approach to using Computer Technology for Education and Training, International Journal of Computer Aided Manufacturing, Vol. 13, pp. 286-297, May-June 2000.
10. Swart, W., Zahorian, S., Lakdawala, V., Leathrum, J., González, O., A Multi-Use Architecture for Technology-Based Delivery of Curricula, Paper presented at 29th Annual ASEE/IEEE Frontiers in Education Conference, Nov 10-13, 1999, San Juan, Puerto Rico, 12c2-12 – 12c2—17.

JAMES F. LEATHRUM, JR.
James Leathrum is an Assistant Professor of Electrical and Computer Engineering at Old Dominion University. His primary research interests are in the areas of modeling and simulation, distributed systems, and formal methods in computer system design. He is also working on projects for technology-enhanced education.

OSCAR R. GONZALEZ
Oscar González is an Associate Professor of Electrical and Computer Engineering at Old Dominion University. His research interests include multivariable system and control theory, robust control system design, nonlinear control systems, and artificial intelligence applications in the control of systems. He is also working on projects for technology-enhanced education.

STEPHEN A. ZAHORIAN
Stephen Zahorian is Professor of Electrical and Computer Engineering at Old Dominion University and chairman of the department. His primary research interests are in the areas of automatic speech recognition and development of speech training aids for the hearing impaired. He is also working on projects for technology-enhanced education.

VISHNU K. LAKDAWALA
Vishnu Lakdawala is an Associate Professor of Electrical and Computer Engineering at Old Dominion University. He is currently the Undergraduate Program Director in Electrical and Computer Engineering. His primary research interests are in the areas of electron attachment in fluorine compounds, breakdown studies in compressed gases and vacuum, material characterization and simulation studies in compound semiconductors, and high power semiconductor switches.