

An Intelligent Tutoring System with an Automated Knowledge Acquisition Mechanism

Dehui Li

Dept of computer and information sciences
Towson University
Towson, Maryland 21204
dli2@students.towson.edu

H Harry Zhou

Dept of computer and information sciences
Towson University
Towson, Maryland 21204
Hzhou@towson.edu

Abstract-The intelligence of E-learning system has become one of regarded topic to public. Using adoptive E-learning system to supplement or replace the classroom is becoming more common in education. To develop an automated knowledge Acquisition intelligent tutoring system is the aim of this paper. Moreover the knowledge Acquisition system (KAS) can improve the education efficiency highly since this system is to automate the knowledge acquisition process, and to perform an analysis based on the students' performance and to select the materials intelligently appropriate to the students' levels and understandings.

Key word: E-learning, Adaptive learning, Knowledge Acquisition, intelligent Tutoring System

I. INTRODUCTION

Nowadays, accompanying the rapid growth of information technology, the intelligence of E-learning systems has become of a surge of interest in the research community. Adaptive e-learning is able to provide adaptive and tailored learning by supporting [1] different learning paths and materials suitable for different students' needs and backgrounds. Many E-learning systems have been implemented on different platforms using different algorithms. The traditional e-learning systems were criticized in terms of their limitations since they always present the same materials and topics to students regardless their backgrounds, learning abilities and understanding levels. Unlike the traditional e-learning systems, intelligent tutoring systems employ a knowledge base which contains subject matter experts' knowledge, teaching strategies, and heuristics. They should be able to select relevant teaching materials dynamically, and therefore choosing different teaching paths, examples and exercises for different students. To solve 'how to transfer knowledge from subject matter to the intelligent e-learning system', we design an intelligent tutoring system with a knowledge acquisition component called KAS that is intended to collect principles, examples, performance indicators, and quality thresholds from the subject matter experts, select an appropriate curriculum for the learner based on students' [2] feedback, and finally generates new study materials appropriate to [4] their levels. The significance of this system is to automate the knowledge acquisition process, and to perform an analysis based on the students' performance and to select the materials intelligently appropriate to the students' levels and understandings.

II. FEATURE

One important component of intelligent E-learning systems is an intelligent knowledge acquisition system capable of gathering knowledge from subject matter experts, such as theories and practices. If the material is abstract, the system will ask for examples and detailed explanations. To apply various teaching strategies, KAS can choose top-down, bottom-up, depth-first and bread-first strategies as appropriate. Our system has organized and collected knowledge and teaching materials in a layered hierarchy implemented in an object oriented environment. It includes the following capabilities and features.

- **User-friendliness**
KAS uses a menu interface with a graphical display to show different level materials and concreteness. There are two modes: a passive mode and an active mode.
- **Dynamics**
KAS helps different subject matter experts to select appropriate exercises and examples dynamically [7].
- **Flexibility**
KAS can adapt the course materials and exercises according to the students' needs and levels [4].
- **Domain-independence**
KAS is reusable for developing curriculums for various new courses and subject areas.
- **Expandability**
KAS is designed in an object oriented environment in such a way that is easy for system administrators to expand and modify without changing its structure.
- **Intelligence**
KAS can remind the subject matter expert of what should be provided at certain points such as common errors associated with a complicated procedure, or specific examples with an abstraction.
- **Reasoning by analogy**
Based on the student's feedback and performance, KAS selects a set of teaching strategies from its accumulated experience, and updates its analogies from its accumulated experiences.

III. SYSTEM DESCRIPTION

KAS is an intelligent tutoring system which includes following the elements: an interface, a knowledge base, an intelligent decision-making mechanism, an inference engine, a database, and an explanation subsystem. The figure1.1 shows the basic structure of KAS.

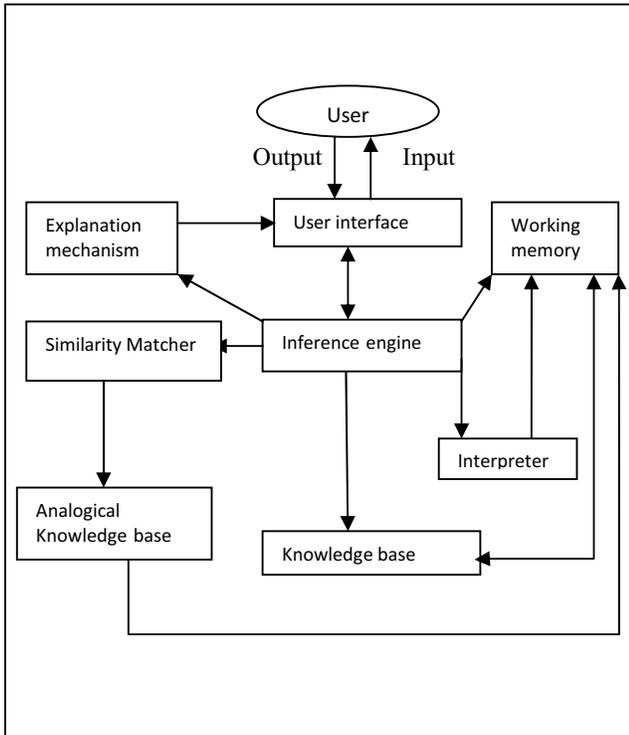


Figure1

In what follows, these components will be explained briefly.

The knowledge base in KAS contains knowledge acquisition strategies and domain-dependent vocabulary. The basic knowledge representation is a frame scheme. A frame is divided into describe elements which is called a 'slot.' Each slot describes as an attribute, which may, in turn, contain one or more facets. Each facet may be the value of an attribute. Another way is a default value that can be used if the slot is empty. An "if-need" facet is also used. If no slot values are given the "if-need" will trigger a procedure that goes out and gets or computes a value. The procedure is known as the procedure invocation. In the context of KAS, a slot may contain a set of examples, or a teaching method in the form of procedure. Frames are excellent for packaging well known or generalized attributes of a complex concept. As frames are linked together in a hierarchy, one frame may inherit the properties of a higher-level frame; the ability of frames to inherit properties makes knowledge storage more compact and permits in depth reasoning.

The interface component controls the reasoning process. The program works with the input data supplied by the subject matter expert, and then organizes them into forms understandable by KAS. Its control strategies implement a combination of forward chaining and backward chaining. The pattern matching function of inference engine compares the input data to precondition part of objects. When a match is found, the matched object sends a message to another object which decides how to organize the knowledge, or what to do next. The inference engine runs the whole process. The two basic functions are inference and control. Inference refers to examine the objects and perform the pattern-matching while control refers to the [3] sequence in which objects are examined. The inference engine asks the subject matter expert for additional input information, such as examples, or explanation if needed. The inference engine fully automates this process and it is totally invisible during the session.

The data base also known as a working memory, it is the portion of the computer's memory set aside for [5] tracking inputs, intermediate data and outputs. The inference engine uses the database like a scratch pad to track what is going on in the system. At the end of a run, the subject matter expert's knowledge is sent to appropriate slots of knowledge structure.

The explanation mechanism explains to the subject matter expert what is needed for a specific exam, how a new concept is connected to the other objects already in the concept hierarchy, and what additional information is required. The expert can also use the explanation mechanism for debugging purposes. During the knowledge acquisition process, it can serve as a way to get feedback on the concept hierarchy construction, enabling the expert to readily test it on practical problems.

In addition to the basic components of an expert system, KAS also has the following features: adaptive ability, learning mechanism, knowledge representations, user interface, and knowledge hierarchy.

A. Adaptability

KAS will not be restricted by a preplanned behavior from a good tutor's instructional efforts. It is very difficult, if not impossible, to build a knowledge acquisition system that collects teaching strategies, examples and domain knowledge from a subject matter expert in a fixed manner. The system is designed in such a way that it can remind the subject matter expert of missing information, such as common errors and necessary examples, and lead the expert to supply all the information and knowledge. Based on the data provided and questions asked by the expert KAS can adaptively decide what questions to ask next and how to organize the knowledge and material collected from expert. Since the system uses several knowledge representation schemes to store factual and strategic knowledge, such as if-then rules, semantic network, conceptual hierarchy, and procedures, it also needs to dynamically select an appropriate form for given information, and make

transformation between two knowledge representations in necessary.

Having collected knowledge from the expert, KAS needs to adaptively refine its knowledge bases as new problems are solved and new problems are experienced. In a learning session, a student's model may need to be modified, as a new concept is introduced or new materials are presented. Over the course of training, many unexpected events may happen which cannot be preplanned at the time of the design. The adaptability allows KAS to handle these events dynamically and makes it applicable in a variety of domains.

B. Learning Mechanism

Learning enables a tutorial system to do the same task or tasks drawn from the same domain more efficiently over time. It distinguishes a truly intelligent tutoring system from traditional computer aid training programs. By fixing a set of tutoring strategies, it leaves no room for improvements over time. To design and construct a general-purpose knowledge acquisition system, it is necessary to augment for the system to refine the course materials over time without being told by the designer. KAS has the ability to learn, to improve, and to tailor its knowledge base during its life cycle, at least to some extent. For example, if the students' approach leads to be a better solution than the one stored in the expert model, the system will adapt the student's approach as its' standard. One method to implement the learning mechanism in the proposed system is through learning by experimentation. It deals only with the teaching strategies and adjusts system parameters that control the trade-offs between conflicting object functions, such as increasing the number of student completing the session, improving their scores on the post-test, and decreasing the time spent by students to go through a training session.

Another form of learning is accomplished by means of feedback and adaptation, which plays an important role in the automated knowledge acquisition process. The feedback from the learner can be used to analyze the problem he or she is confronted with. It also can be used to reveal the student's weakness and estimate the distance between ultimate expert knowledge and the learner's current state of comprehension. For example, the proposed system tries to understand why a student asks a particular question at a particular stage, and whether the question is relevant or not. With the feedback, the curriculum, can be regrouped and rearranged to a form more appropriate to the student. In order to automatically modify tutoring methods, a critic is needed to assign credit or blame to the tutoring methods on the basis of evidence that they succeeded or failed in teaching materials to the student. KAS adapt its tutoring methods to improve itself as a form of learning. By storing a set of tutoring methods with high credits along with the student's performance analysis, it is possible for KAS to model students more accurately. Over time KAS constructs a group of student models of different types based on the

level and common problems, and assigns appropriate teaching strategies to individual students accordingly.

C. Knowledge representation and hierarchy

In the design of KAS a dynamic approach is employed in which topics are selected based on the student's knowledge as well as learning abilities. Some materials may be skipped over if understood, while other topics may be repeated if misinterpreted by student. In order to explain how this mechanism could be implemented by KAS, we need to describe the knowledge representation in which curriculum is encoded. The knowledge representation employed in KAS can be classified into three layers: the knowledge, [5] the goal level and the classification level.

The intermediate layer, called the goal level, consists of conceptual knowledge and goals to be achieved through the means of teaching; the internal structure of this layer is a tree, representing the progressive decomposition of each layer of sub goals into still smaller sub-sub goals. An example is the concept of recursion in the context of Computer Science, which consists of two sub goals: recursive procedure and recursive functions. These sub goals can be broken down into basic components of a recursive function. Such as parameter passing, procedure invocation, and be then decomposed further if necessary and so on, the diagram may become rather complex, but its underlying structure and logic are straightforward. Each sub goal is either a lesson that can be taught independently as a unit, or it may be further decomposed into a series of lessons. This structure is also known as a hierarchy of concepts. If a concept is located at the bottom level of the structure, there are no more examples or further discussions are needed below the current node. Similar materials can be skipped over, and more advance concepts can be presented, which reside at higher levels of concept hierarchy. However, if the student made errors when learning a concept, it may be necessary to introduce or repeat some fundamental concepts which are stored below the current node. It is worthy pointing out that in some applications a goal lattice instead of a fail tree may be needed to represent a cross indexed scheme where a concept can be connected to more than one parent. This type of curriculum knowledge structure supports a number of viewpoints on the goals of instruction. With respect to each viewpoint, one can traverse a subset of curriculum that is a sub concept tree structure. Starting from any specific goal, there are several paths that determine the sequencing of instruction, there are several ways to present the course materials, such as the depth first search, breadth first search, top-down, and bottom -up approached. This structure makes it possible for the proposed system to dynamically select the course and materials, and to change the sequence of instructions in response to this student's feedback and questions.

Thus a teacher can concentrate on important issues and rationales, instead of a step-by-step explanation approach. For a student who needs to have more help, however, the system can traverse the tree structure node by node, presenting every example, exercise, and explanation

attached to the sub tree. Basically, the classification knowledge represents the system's estimates about what students know and what problems they may have. By classifying a student into a particular category, KAS knows which teaching strategies are more efficient, which examples are needed, which materials can be skipped over, which sequence of instruction is appropriate and which topics need to be repeated. In sum, this knowledge structure provides a feasible way to connect the lessons of the curriculum to the knowledge the system is trying to teach. Therefore it is possible for KAS to tailor the course materials to the need of individual students and select exercises to remedy any weakness.

D. The Interface

A graphical interface provides the access for the learner to the system. Before each new concept or subject to be covered, KAS first needs to diagnose the level of student's knowledge and understanding. The interface gives a short quiz to the student. Upon the analysis of the answers provided by the student, KAS chooses a similar student model stored in its knowledge base for an appropriate instruction set and examples. It is believed that similar errors may be caused by the similar deficiencies. Therefore, the instructions followed and examples presented focus these areas for in-depth discussions and explanations. The above process may be repeated until the student fully understood the topic. By doing so, the tutoring system would not follow a pre-determined path and materials to present same teaching materials for all students. As we know, students with different backgrounds are learning at different speeds. For a quick learner, some materials and explanations may be skipped over while for others more examples and exercises may be needed. During previous sessions, different student models have been constructed based on the type of mistakes and incorrect responses to certain types of questions. Each model consists of two parts: attribute and plan sections. In the attribute section, the descriptors of errors made by students are stored representing the type of deficiency this group of students may have in common. In the plan section, a set of examples, exercises, and explanations are stored focusing specifically to address the errors. Once a student's deficiency is identified by KAS, the relevant teaching materials are retrieved from

the model and are presented to the student. By doing so, the intelligent tutoring system is able to tailor the teaching material for each individual student, and to achieve efficient and dynamical teaching.

IV. CONCLUSIONS

By merging the existing techniques and systems developed by the authors and other researchers, and by exploring the new research frontiers, it is hoped to make contributions to the development of automated knowledge acquisition systems that could lessen the bottleneck problem existing between ITSs and subject matter experts. It is believed that any success and progress in this direction will shed light on some basic research issues, such as analogical learning, adaptive interactions, as well as have a significant impact on the development of intelligent tutoring systems.

Reference:

- [1] Manju Bhaskar, Minu M Das (2010) "Genetic Algorithm Based Adaptive Learning Scheme Generation for Context Aware E-learning" (IJCE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1271-1279
- [2] Samir ALLACH, Mohammad ESSAIDI "Towards optimizing learning paths in a system E-learning adaptive application of ant algorithm (ACO)" (JCES) International Journal of Computer Engineering Science Volume1
- [3] Musa, A. and Armess, A. (2010) "Investigation on the effectiveness of Learning's Sequence in E-Learning Environment", The 2010 International Conference on Educational and Information Technology (ICEIT2010), 17-19 September 2010 in Chongqing, China.
- [4] L. Zaitseva, C. Boule, "Student Models in Computer Based Education," Proceedings of the third IEEE International conference on Advanced Learning Technologies, 2003, 674-676
- [5] Jeremic, Z., Jovanovic, J., & Gasevic, D. (2009). "Evaluating an Intelligent Tutoring System for Design- Patterns." the DEPTHs Experience. Educational Technology & Society, 12 (2), 111-130.
- [6] S. Azough, M. Bellafkih, EL H. Bouyakhf (2010) "Adaptive E-learning using Genetic Algorithms" IJCSNS International Journal of Computer Science and Network Security, VOL. 10 No. 7, July 2010
- [7] John A. Kaliski, Queen E. Booker, Paul L. Schumann "An Architecture For Dynamic E-Learning Environments Based On Student Activity And Learning Styles." Business Education & Accreditation, Volume 4, Number 2. 2012