N90-27320

A STRUCTURE FOR MATURING INTELLIGENT TUTORING SYSTEM STUDENT MODELS

Dr. Willard M. Holmes
Systems Simulation and Development Directorate
Research, Development, and Engineering Center
U.S. Army Missile Command
Redstone Arsenal, Alabama 35898-5252
Phone 876-1048

ABSTRACT

In this paper, a special structure is examined for evolving a "Detached" model of the user of an intelligent tutoring system. Tutoring is used here in the context of education and training devices. A "Detached" approach to populating the student model data structure is examined in the context of the need for time dependent reasoning about what the student knows about a particular concept in the domain of interest. This approach, to generating a data structure for the student model, allows an inference engine separate from the tutoring strategy determination to be used. This methodology has advantages in environments requiring real-time operation.

INTRODUCTION

During the past decade, a considerable increase in research on Intelligent Tutoring Systems (ITS) has resulted in an expanded body of knowledge about computer based tutoring systems. ITS are sharply contrasted with what is traditionally identified as Computer Aided Instructions (CAI). Early research on the distinguishing characteristics of ITS and differences compared with CAI are reported in a reference text "Intelligent Tutoring Systems" edited by Sleeman and Brown (1982)[7]. Wolf and McDonald (1984)[9] emphasizes the importance of student modeling in developing an effective tutoring system. A general state of technology development in the emerging technology of ITS is reported on by Clancy (1987)[1], Wenger (1987)[8], Kearsley (1987)[5], Polson and Richardson (1988)[6].

In addition to the man-machine interface, the classic model of an ITS includes a teaching module, an expert problem solving module, and a student model. A student model is an essential component of ITS. However, a student model in general cannot be developed entirely independent of the domain in which the model will be used. The vast majority of student model development efforts focus on the "Classic" tutoring problem, i.e., duplicating the tutoring function that takes place in a classroom. Another area of interest in the use of tutoring and student models is related to training and job aids. A mental model for identifying some differences between "Classic" instructional strategies, training, and job aids are reported on by Harman and King (1985)[2].

The use of a student model in context of a specific application domain is described by Holmes (1988)[3] and Holmes and Chamberlain (1988)[4]. For purposes here, the student model is defined as that component of an ITS that collects student model performance information to be used to make inferences about what the student knows and does not know about a particular concept or required training task. Before the student model can be used to draw conclusions about the state of knowledge possessed by the student, the model must first be initialized or have results available from previous tutoring operations. More precisely, the student model must have a specified structure and a defined process for populating the structure. With the populated structure, inferences can be made about the knowledge state of the student.

By using an ITS in the context of a Training System (TS), it can be identified as an Intelligent Tutoring-Training System (ITTS). The task to be presented to the student in an ITTS application is similar in nature to the basic principles that would be used for an ITS for the "Classic" knowledge tutoring problem associated with classroom settings. In the ITTS operation, simulation systems are frequently included to support the exercise of both knowledge and skills in the tutoring operation.

HIERARCHICAL DECOMPOSITION OF TUTORING TASK

Given that a main level concept with specific performance objectives has been identified as an element to be used in a tutoring operation. A series of steps must be completed before the tutoring function can be implemented. An initial step is to perform hierarchical decomposition of the main concept into subconcepts. The decomposition continues until desired fidelity level is acquired. The fidelity of the decomposed task is related to the number of levels in the hierarchy. The lowest level subtask in the hierarchy is defined here as the component subtask. Skills and knowledge associated with the component subtask is identified as the primitives of the task. This decomposition process is typically an element that is part of a total task analysis effort. In conjunction with the task analysis is the skill analysis to identify requisite skill associated with the task elements. A conceptual model for identifying the task analysis components is shown in Figure 1.

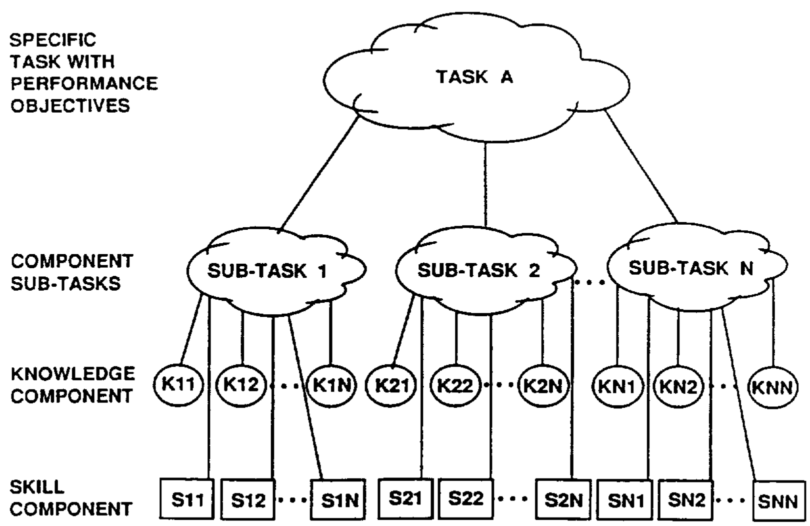

Figure 1. Conceptual model for identifying task analysis components.

The hierarchy of skill components indicated in Figure 1 is a simplification of the many subtask levels that can be associated with a task. A major point here is to focus on the desirability of having associated with each subtask a concept that is an element in the major task. The knowledge and skill components associated with the subtask is the knowledge necessary to master the concept and the skill to demonstrate the operation associated with the subtask. Knowledge is defined here to include what the student or operator needs to know (principles, concepts, facts, etc.) about the subtask to be accomplished. A skill is defined as having the requisite knowledge and the ability to apply that knowledge effectively.

SKILL BASED TASK ANALYSIS

An example emphasizing the use of information at the component subtask level with associated knowledge component and skill component is instructive. Consider this over simplified example of teaching a student how to fly a small airplane as indicated in the following table.

The observed performance with resulting conclusion and recommendation is typical of that made by a human tutor. The knowledge component is information the student can obtain in classroom sessions, books, and discussion with an experienced pilot. The measure of the student's knowledge can be a series of questions. The skill component is developed and tested either with a training device or the actual airplane. The performance measure can be an observation and measurement of action taken in response to a given stimuli. As can be observed, the same knowledge and skill component can appear in more than one subtask, i.e., adjust controls in ST.-1 and ST.-2. This is the same skill but the student must have the knowledge of the context of using that skill, i.e., in landing or stall situation. Hence, the ability to use the knowledge correctly.

The expert, in expressing his rule-of-thumb, may use terms and expressions not used at the knowledge component or the skill component level. However, the student's action can be compared with the expert's at the subtask level. Tutoring operation to improve deficiencies can use student performance at the subtask level and the related information at the knowledge component and skill component level.

STRUCTURE FOR STUDENT MODEL DATA GENERATION

For purposes of prototype model development, the structure shown in Figure 2 will be used to identify a procedure for developing student model data. As indicated, the basic structure includes a main concept, one or more sun-concepts, and primitive element. The primitive elements have context sensitive Primary (P) and Alternate (A) question associated with each primitive.

TABLE. Learn to fly a small airplane.

COMPONMENT SUBTASKS:

SUBTASK.1	SUBTASK.2	SUBTASK.3
ST.1 Learn how to land	ST.2 Learn how to take off	ST.3 Learn how to handle stalls
KNOWLEDGE COMPONENTS		
Head winds Cross winds Flight path angle Landing speed	Plane load Air temperature Take off speed Aerodynamic lift	Get nose down Increase speed Sufficient speed Control settings
SKILL COMPONENTS		
Adjust controls Observe airport wind indicator Compare approach Angle with horizon Adjust air speed	Set throttle Observe cross wind indicator Changes controls with correct air speed	Adjust controls Observe air speed Level off

OBSERVED MEASURE OF PERFORMANCE FOR SUBTASK.1

Approach angle too high, hard landing
Landing speed too high, ran off runway

CONCLUSION

Student and plane survived, but the student need more practice.

RECOMMENDATIONS

The student engage in extended practice session of touch-and-go landings with emphasis on control surface adjustment and speed adjustments.

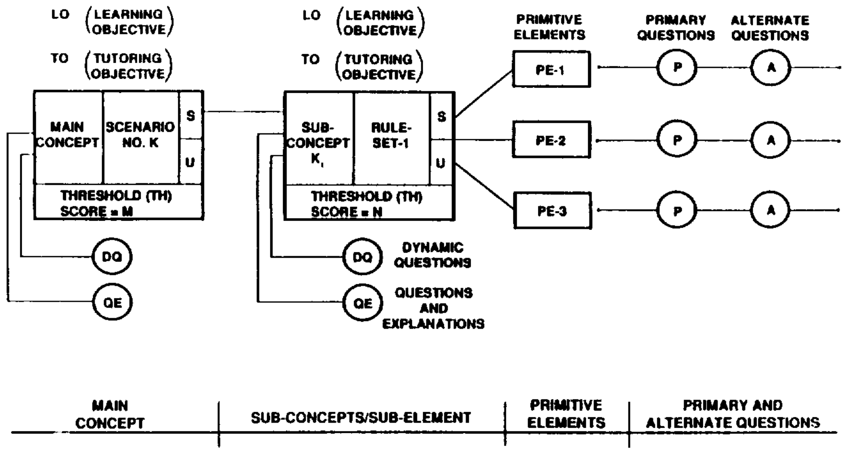

Figure 2. Basic structure for developing student model information.

The main concept is considered to be a particular scenario operation. A Learning Objective (LO) is identified for the main concept. The LO will be dependent on concepts included in the scenario. The Tutoring Objective (TO) will be dependent on the particular student being tutored, i.e., student model information and capabilities required to solve the scenario. As used here, a TO is related to the process of establishing the number of chunks of knowledge to be presented to the student. In turn, a LO is related to the process of tesing to determine if sufficient knowledge has been mastered. Obtaining student model information at the main concept level may include presenting the student with a set of Dynamic Questions (DQ) involving both knowledge and skill or presenting static Question with Explanation (QE) as required.

The student may not be required to master 100 percent of the knowledge and concepts contained in this main event (scenario) Rule Set (RS) before advancing to the next main concept (scenario). A Threshold (TH) of performance is established for the main concept. An indication that the LO has been Satisfied (S) or Unsatisfied (U) is indicated by setting the appropriate performance indicator. The TO can be satisified even if the LO was not satisified. Results of the student's performance at the main concept level is added to the student model knowledge base.

Entering into a particular subconcept operation can be accomplished by one of three approaches. The learning objectives associated with the main concept was not satisifed and the tutoring strategy directed that a subconcept of the main concept be explored, (top down approach). Second, the tutoring strategy and student model contents indicate that the bottom line component subconcept be explored before advancing to a higher level concept (bottoms up approach). The bottom line subconcept level is defined as the level directly connected to the primitive elements.

The third approach to subconcept operation is associated with paths that includes several subconcept levels between the main concept and the primitive level. Under these conditions, the tutoring strategy can require that operations proceed to a particular subconcept to satisfy certain TO (arbitrary approach). This approach would be applicable to an expansion of the basic structure as shown in Figure 2. The subconcept levels directly associated with the primitive elements are identified as the subelement level.

The student model contains results of the student performance at all subtask levels, including the component subtask level. Any interaction with the primitive elements during a TO is not recorded in the student model. This requirement is tied to the fact that knowledge about the student's performance consist of two parts: the student's performance during Past Tutoring Efforts (PTE); and the students performance in the Present Tutoring Operation (PTO). The student's performance is not considered to be a past performance until the PTO for a subtask is satisifed.

Consider the option that while achieving a TO it is necessary to enter the subelement level of operation. Also, consider while satisfying the LO of a particular subelement it is necessary to interact with the associated primitive elements. The P question and A questions are used to inform the student about the characteristics of the primitive elements. Primitive element level interaction with the student continues until the threshold level of that particular subelement is achieved. When the performance threshold level of the subelement is achieved, the results at the subelement level is recorded in the student model. This gives information on the student's performance in relation to particular LO associated with the subelement and not in relation to the primitive elements. At this point, the results are based on past performance since the LO has been achieved. The interaction with the primitive elements occurred before the TO was achieved, i.e., in a present scenario operation mode.

This requirement, that the threshold level of the subelement be satisfied before continuing the TO with higher level concepts, can be used to establish lower bounds on the tutoring and training operation. An ideal structure would have a small number of primitive elements associated with each subelement, i.e., three. If the threshold performance level is set at 100 percent and the student cannot achieve the LO of the component subelement, then the student should receive training of a more fundamental nature than the particular ITTS can provide.

However, other options exist if the performance threshold level of the subelements are set at some value other than 100 percent. No record is kept of the order the PEs are presented to the student during the first tutoring session using the particular subelement. Since a flag was set indicating results of the last LO, at least one pass has been made using the subelement. In the event a second pass is required to meet a TO, the order of presenting the PEs on a second pass through can be reversed to make it interesting for the student. Shown in Figure 3 is a conceptual model of an expanded structure for student model data development.

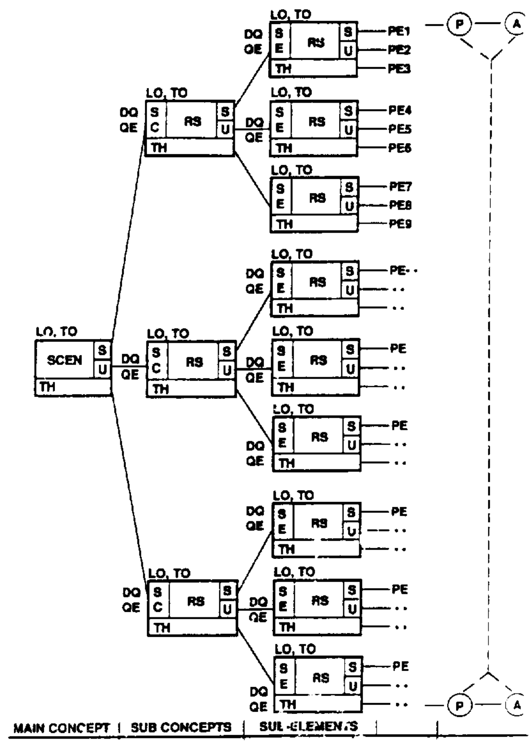

Figure 3. Expanded structure for student model development.

CONCLUSION

An initial implementation on the prototype structure, as described here, is effective in providing required information to a student model for a special application of tutoring systems. The implementation of the structure, described here, involved an initial prototype. The next step is to investigate the effectiveness of applying the process to larger tasks with an increased number of subtasks.

REFERENCES

1. Clancy, William J., Knowledge-Based Tutoring: The GUIDON Program, The MIT Press, Cambridge, MA, (1987).

2. Harmon, Paul, and King, David, Artificial Intelligence in Business, John Wiley and Son, Inc., New York, NY, (1985).

3. Holmes, Willard M., and Chamberlain, David, "A Scenario Generator for an Intelligent Tutoring Device used in Maintaining Operator Skill Levels," Proceedings of the 1988 Summer Simulation Conference, Seattle, WA, pp. 613-619, July 1988.

4. Holmes, Willard M., "A Structure for Developing a Domain Specific Intelligent Tutoring System for Maintaining Proficient Skill Levels for Weapon System Operation," 2nd Annual Conference on Advancement in Training Technology, National Science Center for Communications and Electronics, Augusta, GA, 26-18 April 1988.

5. Kearsley, Greg, Artificial Intelligence and Instructions, Addison-Wesley Publishing Company, Menlo Park, CA (1987).

6. Polson, Martha C., and Richardson, Jeffrey J., (Eds.) Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates, Inc., (1988).

7. Sleeman, D., and Drown, J. S., (Eds.) Intelligent Tutoring Systems, Academic Press, Inc., New York, NY, (1982).

8. Wenger, Etienne, Artificial Intelligence and Tutoring Systems, Morgan Kaufman Publishers, Inc., 95 First Street, Los Alto, CA, (1987).

9. Wolf, Beverly, and McDonald, David D., "Building a Computer Tutor: Design Issues," Computer, IEEE Publication, pp. 67-73, September 1984.