From: Proceedings of the Twelfth International FLAIRS Conference. Copyright © 1999, AAAI (www.aaai.org). All rights reserved.

A Case-Based Reasoning Approach to Internet Intelligent Tutoring Systems (ITS) and ITS Authoring

Richard Stottler & Sowmya Ramachandran

Stottler Henke Associates, Inc.
1660 So. Amphlett Blvd. Ste. 350
San Mateo, CA 94402
Phone: (650) 655-7242 Fax: (650) 655-7243
stottler@shai.com, sowmya@shai.com

Abstract

This paper describes three aspects of our ITS research and development project currently underway for the Air Force.[1] The purpose of our project is to develop both a general ITS authoring tool and a specific ITS to train AWACS Weapons Directors. Our approach is innovative in three ways. First, instruction, assessment and remediation all occur primarily in the context of a simulated scenario, or case, with the student performing actions (primarily cognitive) in as close to an operational environment as possible. The second innovation is the use of the Internet to provide distributed tutoring through a "loose confederation" of ITSs. The third innovation is the development of an authoring tool, for use by non-programmers, to enter both the domain and pedagogical knowledge, required for the ITS, which is then used to automatically create the corresponding ITS. This provides an ITS authoring capability.

Use of Cases in ITS Development

The original motivation for our use of cases within an ITS was from the perspective of Case-Based Instruction. That is the notion that students learn best from examples (scenarios) and that their abilities should be tested within the context of scenarios which were as close to those that they would experience in an operational environment as possible. As we expanded our ITS work, it became apparant that cases are useful for other reasons as well. Stottler Henke Associates, Inc. (SHAI) works within many domains in which it is considered impossible or impractical to build a computational system which can perform at the level of the human expert. This prevents us from using the strategy employed in the development of many ITSs that of developing an explicit expert model which can perform at the required level of expertise. Thus cases became important that as a means of determining if a student's actions in a scenario were correct, in domains where a general system to produce these correct actions was unattainable. A third, related use of cases is as a basis for the representation of the domain expert's knowledge, which is discussed immediately below.

Cases for Knowledge Representation

There are several useful knowledge elicitation strategies related to Cases. These are amply described in [Stottler 1988]. Recently cognitive psychologists have begun to utilize many of the same knowledge elicitation techniques and describe them under the term Cognitive Task Analysis [Klein 1997]. These methods seek to elicit knowledge by having an expert explain his actions in a particular scenario, and the reasoning behind those actions. In many complex domains, this is the only way that experts can explain their thought processes. The knowledge and reasoning processes are too complex to explain generally; they can only be described in the context of specific circumstances. In these types of domains, using cases as a basis for representing expert knowledge is clearly called for.

The expert is asked to present an actual specific problem and its solution, with an explanation of the steps required to produce the solution. The explanation refers to principles or concepts underlying the problem's solution process. Thus, the principles serve as the organizational structure of the knowledge, and the scenarios serve to illustrate concepts. Because a case-by-case approach to knowledge gathering is highly intuitive, authoring of a training course is greatly simplified and requires no special computer training. Further, maintenance of the ITS and addition of updated course material is accomplished primarily through addition of new cases. As described in the next subsection, cases are very naturally used in instruction. They also allow for a very rich representation, since additional knowledge can be easily attached to the case. This additional knowledge can even be in the form of natural language or multi-media simulations, since they can be presented to the student before, during, or after the scenario (e.g. they can be used to debrief the student after an exercise).

[1] See Air Force Contract Number: F41624-98-C-5007

In our ITSs each case (or scenario) includes (1) a multi-media description of the problem, which may evolve over time (as in tactical scenarios); (2) a description of the correct actions to take, possibly including order-independent, optional, and alternative steps; (3) multi-media explanations of why these steps are correct; (4) the list of methods which determine whether the steps have been correctly executed by the student; and (5) the list of principles required to know the correct action to take, typically extracted from the explanations that accompany the solution steps.

Case-based Instruction

In complex domains, instruction is often complicated by the need for the student to master a variety of concepts and to apply them in unique situations and in different sequences. In these kinds of domains, the student must develop a competence not only in the relevant facts and skills, but also an understanding of the concepts underlying these procedures. Given these requirements, the use of cases, or scenarios, facilitates the development of the required cognitive depth and flexibility. For example, AWACS Weapons Directors (WDs) have to understand the evolving air battle, then give situation updates and make recommendations to the appropriate pilots, without overwhelming them with information. AWACS instructors have learned that playing many tactical scenarios is extremely important in training weapons directors.

Our ITSs use an extensive case-base of scenarios as exercises and examples to teach students. Research has revealed that students learn best when they are presented with examples of problem-solving knowledge, and when they are required to apply the knowledge to real problem-solving situations. The case-base of examples and exercises captures such realistic problem-solving situations and presents them to the student, typically within a simulation. The student is required to interactively solve the problems, thus giving him an opportunity to practice the necessary skills as well as to reveal areas of knowledge deficiency. The ITS monitors the student as he performs these simulations, diagnoses the strengths and weaknesses of his knowledge based on his performance, updates the internal model of his knowledge, and tailors instruction in order to correct the weaknesses. Each action that the student was to perform in the specific case was specified by an expert along with an explanation as to why that action is appropriate. That explanation references the principles that the student must know, at least intuitively, to be able to decide to perform the correct action. Thus, if a student's action is incorrect, the ITS can hypothesize a weakness in one of the principles listed in the action's explanation. Each action of each exercise allows the ITS to gather further evidence. This evidence is used to determine the principles, or knowledge, in which the student is weak.

Depending on its diagnosis of the student, the ITS may display an example relevant to the principles being taught at the time, along with the expert's correct actions. In simulated scenarios, the expert's actions "play" the scenario for the student. Since each action also includes an explanation as to why that action is appropriate in the scenario, this information is available for explanations to the students. It may test the student with an exercise which uses principles that it believes he has learned. It may debrief the student on the mistakes of his last exercise. Or, it may formulate a remedial course of instruction, based on the deficiencies in the student's current mental model. These remediations may take the form of examples or general topic information, followed by exercises to test the effectiveness of the remediations.

In this project and other ITS projects for the military, we have found that military instructors and trainers heavily favor the use of tactical simulations and believe that the best trained officer will be the one who has experienced the most tactical scenarios. Similarly, we have found in our work with astronauts at NASA, that the best trained astronaut will also be the one who has experienced the largest number and variety of scenarios.

Cases for Correct Action Determination

The most difficult and domain dependent aspect of the process described above (after the simulation itself) is the determination of the correctness or incorrectness of a student's action. Since we work in domains where it is impractical to build a general expert system to produce the correct actions, instead we store the expert's knowledge of the correct actions specific to a scenario within the scenario itself. This knowledge typically takes one of three forms, based on the domain and the ability of the student to alter the flow of the scenario in unexpected or multiple ways. The simplest representation lists the correct actions at the appropriate time in the scenario. Obviously this will only be applicable if the flow of the scenario is unaltered by actions of the student or if at each mistake, the student is immediately corrected, and thus the scenario's timeline is restored. For each scenario, methods are required for comparing these correct actions to the actual actions produced by the student. These methods may also be able to assess which principles associated with a particular action the student knows and which ones he doesn't based on a whole or partially correct action. For example, in some AWACS Weapon Director (WD) scenarios, the WDs are supposed to advise rather than command. Thus the scenarios can be structured such that the simulated pilots ignore WD

mistakes, and the scenario timeline procedes unaltered. The WD actions are the advice, specific utterances made to specific pilots over the simulated radio, usually less than 20 words each. The correct actions are the utterances of expert WDs, previously recorded while they played the scenario. The software methods to compare the correct actions to the student's actions must convert each to a text representation. The WDs, according to their orders, are supposed to use a specific grammar. This allows the text to be parsed and compared piece by piece. The software methods can then assign knowledge of principles based on subparts of the student's utterance. Some principles, such as "give the most important information first," actually span multiple actions, as well.

Of course these types of scripted scenarios preclude one of the most important learning opportunities - for students to see the results of their own mistakes. Mistakes a WD makes in real missions can easily cause loss of life, including his own. So there is a strong desire to use more flexible and dynamic simulations and scenarios, where a student's actions can radically affect the outcome. Since these simulations are typically continuous, there is an infinite number of variations that different students can create. In fact, in these types of situations the same scenario never plays exactly the same way twice, since minor timing differences of student actions affect the precise trajectories of the simulated players. Clearly, listing the correct action at the appropriate time, based on the way the expert played the scenario is inappropriate, since when the student plays the same scenario, his timelime will diverge from the expert's, often in radically different ways. For example, a particular scenario may dictate that the student remain covert while gathering information. If he understands how to do this, the enemy may never detect his existence, and thus never attack him. However, a student who does not understand the principles of covertness may turn on his active sensors, be detected by the enemy, and thus come under attack. At this point he may correctly assess the need for and execute several self-defense actions. These actions were not required of the expert or of other students in the same scenario who performed the information gathering tasks in the correct, covert way. Yet, they are entirely appropriate for the situation in which the student finds himself, and not only should they not be considered incorrect, but he should also get credit for understanding the appropriate self-defense principles.

The solution is the second of the three forms of knowledge. We attach to each case, one or more expert systems. Each expert system is designed to handle a specific type of condition (such as missile attack) which might arise. Each e.s. is activated by its own set of preconditions. These are much easier to develop in a case specific way, than in the general case. Many can also be re-used across scenarios, with little or no modification. In other efforts SHAI has developed techniques to quickly develop expert systems in military domains for the representation and execution of tactical knowledge to produce tactically correct actions [Stottler 1996]. Methods are still required to compare the correct actions to the student's actions. For example, it may be that the student should respond to a particular kind of attack with a set of self-defense actions within 30 seconds. If nothing else, the method which compares the student's actions to the correct actions must allow for this slight time variation. Typically there may be other parameters with allowable variations and several appropriate order-independent actions as well.

The third form in which knowledge of correct actions may be stored and used is in situations where the system in no way can produce the correct or all the possible correct actions but for which the knowledge exists, within the context of a scenario, to evaluate the appropriateness of the student's action. For example, to refine the location of an enemy platform, an aircraft may be sent to a general area. To keep the aircraft's home platform location unknown, it should take an indirect route to the area. There may be several factors to consider when determining an appropriate route, many of which may be considered commonsensical or at least not part of the course the ITS is teaching. The ITS may not include the knowledge required to generate a good route. Furthermore, there may be a very large number of acceptable routes. But, for the purposes of making sure that the student understands the concept of taking an indirect route to the target area, it is fairly easy to devise a simple calculation to check that the route was indirect.

Distributed, Cooperative ITSs via the Internet

To facilitate cooperation among ITSs, our project incorporates the concept of a "loose confederation of ITSs". The inspiration for this original idea came from the world wide web and its current usage. Web pages are developed and maintained by a haphazard, loose confederation of individuals. People browse one web page, which may have links to other web pages, which they may follow and visit, even though the authors of the separate pages are probably unknown to each other. Further, search engines exist to help guide end-users to pages which may be of interest. Our concept for ITSs follows naturally from this model. The concept is a loose confederation of Internet ITSs (which know about each other) maintained by individuals in geographically diverse locations who have little or no interaction. These ITSs would tend to be maintained by people with an interest in education - teachers, university and college professors, and other organizations with an interest in education.

ITSs can coordinate their efforts in the training of a particular individual. The ITSs notify each other of their existence and training capabilities. One ITS can make specific training requests of another, remote ITS, passing it the trainee's current student model and training needs. A user interacts with one ITS. When it determines that he lacks knowledge in a related field, handled by another ITS, it sends him there, and so on. When he switches from one ITS to another, his student model is sent to the other ITS so it instantly knows the state of his knowledge in related fields and his learning style preferences.

This idea is a completely new paradigm for ITSs and could revolutionize education. ITS Authors don't have to duplicate each other's work but can take advantage of it. Instead of developing ITSs with all related and prerequisite knowledge, authors only have to design systems in their primary field of interest, and let others take care of related fields. Every student has equal access to all ITSs and can make use of the best ones in each field, without having to know about them explicitly: the system guides them there and passes along the model of the student's knowledge and learning style preferences. The ITSs are universally accessible to students. That is, if an instructor authors an ITS, he knows it will get used without having to promote its existence. (i.e. bi-directional universal education). Authors can concentrate their time on what they want to teach instead of other subjects or access and promotional concerns.

For example, consider four related ITSs - Wave-Theory, Sound Wave Theory, Electro-Magnetism (EM), and Calculus. Suppose the student starts out interacting with the EM Tutor and it determines that the student is weak in Wave Theory. It sends him to the wave theory ITS, passing along his student model. While interacting with the wave theory ITS, the system determines he is weak in requisite calculus subjects, so it sends him to the calculus tutor, with his student model and a list of what the Calculus ITS needs to teach him (he only needs a portion of the calculus course as a prerequisite for the wave theory course). After mastering the required calculus subjects, he returns to the Wave course and then back to the Electro-Magnetism course. Later, he is assigned to learn the Sound Wave course. His student model is passed to it, and it determines that he already knows a lot of wave theory (from the Electro-Magnetism course), so these topics can be skipped. Since he has also interacted with the Calculus tutor (since he needed to, implying a lack of Calculus knowledge), anything not taught him while there, he's probably not covered before. If a few additional calculus topics are required, the ITS sends him back to the calculus tutor to focus on those topics. The calculus tutor itself might be a loose confederation of ITSs which each address different (or possibly overlapping) topics.

We have already designed and implemented a scheme which would allow the required communication between ITSs. In our concept, a central ITS server exists as shown in the figure below and keeps a master list of all ITSs, their capabilities, and a global list of principles. The ITS authoring tool running locally on the author's machine is able to get information from and transfer information back to the central ITS server enabling the author to send course updates and the instructor to get student models. Finally, the cooperative ITSs are able to send student models to each, launch each other, and get information about each other.

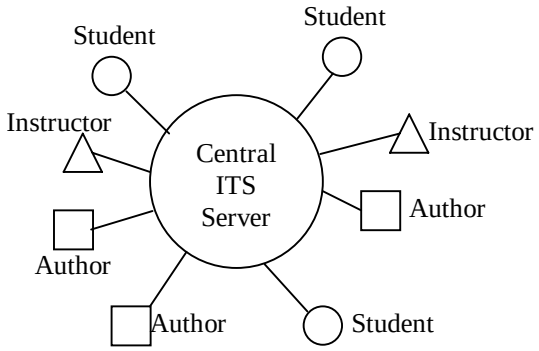

Figure 1, ITS Server

An author creates the ITS at one site, in consultation with the server and sends it the finished ITS (and any updates later). The student receives at his site the ITS (and updates) from the ITS Server, and his student model is transferred to the ITS Server. It can therefore keep a master list of all students. An instructor can access the student models of his students from his own site. Since the local ITS that is running either as a web page or on the student's local machine is delivered from the ITS Server, it can automatically know of related ITSs and the principles they teach and, when required, request these related ITSs from the ITS Server to tutor the current student. SHAI has already implemented the required Internet communication software and currently maintains the ITS Server. Since the ITS authoring tool is still under development, ITSs have not as yet been developed to reside on this server.

Perhaps the major issue that will be faced is in the area of ontology. Each ITS uses its own local language to describe the principles it teaches and which are illustrated and/or required in its scenarios. The central ITS maintains a global hierarchy of the principles

taught by each ITS using a global ontology and maintains a table which translates the names of the principles in each local ITS's ontology to the names of the principles in the global ontology. It remains to be seen if this relatively simple scheme will be adequate, whether different instructors not in communication with each other can cooperate through it, and how much human intervention may be required to "clean up" the global hierarchy which necessarily results from entries made from disparate fields and instructors. The initial assumption is that all ITSs are authored with the same Internet ITS authoring tool, but this constraint can be relaxed later. All that is really required is that each ITS can describe what it teaches in terms of a (possibly flat) principle hierarchy, that these can be linked to principles in the global hierarchy that is maintained by the ITS Server, and that the ITS can describe how and when it should be initiated.

ITS Authoring Tool

There are six kinds of knowledge that the must be entered by the domain and/or instructional expert to create an ITS for a specific domain. These are the case base of scenarios to be used as examples and exercises, the hierarchy of principles referenced from those scenarios, multi-media descriptions which explain each principle, knowledge used to asses the correctness of student actions, knowledge used to assess a student's mastery of a principle given the history of his performance in relation to that principle, and pedagogical knowledge. Methods to enter the scenarios tend to be very domain specific and closely tied to the simulation. For tactical scenarios, we typically employ graphical editors based on an electronic map and intelligent tactical knowledge entry techniques not particularly to ITS concerns. The principle hierarchy is entered through a simple tree-based graphical editor, with opportunities to browse the global principle hierarchy and make desired connections between nodes in the local hierarchy and nodes in the global one. The authors can also add (but not change) nodes in the global hierarchy. The multi-media descriptions of principles are entered using commercial multi-media authoring tools, such as Macromedia Director. The different ways to represent the knowledge to determine student action correctness was discussed in a previous section. While the means of entry tends to be tied to the type of domain, we have found that finite state machines are often useful. In the following paragraphs we discuss the capabilities for entering mastery assessment and pedagogical knowledge.

Representation and entry of the knowledge to assess principle mastery, given a history of actions related to it, is one of the simplest aspects of the authoring tool. The author specifies and names the levels of mastery. For example, those might be novice, intermediate, and expert. For any principle in the hierarchy, he then defines the conditions that must be met to attain each level of mastery. These conditions typically define the percentage of correct usage of a principle from the last N actions using the principle in the last M scenarios in a specified time period. The required parameters are simply entered using a fill-in-the-blank format. Which principles the mastery level applies to is determined by which principle node the author selected in the principle hierarchy editor. The mastery assessment definitions defined at a higher level in the hierarchy are inherited by all of its subprinciples unless over-ridden with a more local definition.

More complicated is the pedagogical knowledge. A somewhat simplified description follows. The authoring tool allows different instructional methods to be defined for different types of students (based on background and principle mastery) and different regions of the principle hierarchy. Aspects of an instructional method include degree of instructional support; degree of student control; how much instructional material to present; what kinds of examples to show, and how many; what kinds of exercises to present, and how many; type and timing of debriefing; remediation, and exercise selection.

Perhaps the greatest challenge in designing the functionality and capabilities of the ITS authoring tool is maintaining the proper balance between flexibility/power and usability. By designing a lot of flexibility in the instructional method specification, many inputs are required. Our design philosophy is based on the assumption that domain knowledge experts (a.k.a. Subject Matter Experts) are more readily available than pedagogical experts and that pedagogical knowledge can be generalized over domains. We therefore have made pedagogical knowledge specification easy by having the authoring tool intelligently select default specifications that an author can choose to over-ride. We are developing a case base of instructional techniques so that when some preliminary information about the domain and types of students is entered, the system selects the most appropriate default instructional techniques for each type of student and principle. A Subject Matter Expert is able to generate an ITS by just specifying domain-specific knowledge (principles, scenarios, pre-tests/post-tests scenarios), and using default specifications for pedagogical knowledge.

A preliminary version of the authoring tool which allows input of the various types of domain-independent knowledge required for the ITS, especially the pedagogical knowledge, now exists. We will soon have a version which uses this knowledge to create an ITS. We would be very interested in getting feedback from other ITS developers as to the default choices we

have made, since we are still early enough in the project to incorporate feedback.

Current and Future Work

We have applied the case-based ITS approach in a number of domains. We have developed an operational system for the US Navy's Surface Warfare Officer's School (SWOS), which they are currently using. We have developed prototype case-based ITSs for tutoring NASA astronauts in experiment procedures and the science which underlies them, for training AWACS Weapons Directors controlling Air battles, and for training sonar technicians to find submarine signatures in sonar images. These prototypes monitor student actions in scenarios, derive a model of the student's knowledge, select the most appropriate examples for instruction and exercises for testing, and select the topics that address the areas that the student doesn't understand. They also provide a facility for the student to browse those topics himself.

We are now developing full-scale ITSs for each of these three areas. In each case, the domain experts and instructors agreed completely with the case-based approach to instruction and with our ITS designs and prototype implementations in particular. In 1999, we will get more quantitative results as the full-scale ITSs are used. We are also developing an ITS authoring tool which attempts to unify these diverse domains. We will then be able to gauge the degree to which instructors can create ITSs, without programming or knowledge engineering assistance.

We have implemented the software to pass the needed information from ITS to ITS across the Internet, as well as to a central repository. We currently are maintaining an Internet ITS Server, accessible from our web page, http://www.shai.com. In 1999, we will begin to field distributed, cooperative ITSs, developed using the ITS Authoring tool currently under development.

References

Aleven, V. and Ashley, K.D. 1992. Automated generation of examples for a tutorial in case-based argumentation. In *Intelligent Tutoring Systems '92 Proceedings*. Montreal, Canada.

Anderson, J.R. and Reiser, B.J. 1985. The LISP tutor. *Byte*, vol. 10. no. 4, pp 159-175.

Klein, Gary and Zsambok, Caroline E., A. eds. 1997. *Naturalistic Decision Making*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Kolodner, Janet L., A. eds. 1993. *Case-based Reasoning*. San Mateo, CA.: Morgan Kaufmann Publishers.

McArthur, D., Lewis, M., and Bishay, M. 1993. The Roles of Artificial Intelligence in Education: Current Progress and Future Prospects, DRU-472-NSF, RAND Education, Draft report.

Regian J. W., & Shute, V.J. 1994. "Evaluating Intelligent Tutoring Systems", In E. L. Baker and H. F. O'Neil, *Technology assessment in education and training*, Hillsdale, NJ: Lawrence Erlbaum.

Stottler, Richard H., and Parekh, Sujay S., *AI Techniques for Reusable Tactics Expert Systems*, Stottler Henke and Associates, Inc., 107-Tactics FR, November 1996.

Stottler, Richard H., *Electric Power Research Institute Knowledge Acquisition Workshop Handbook*, Science Applications International Corporation, March 1988.