Case-Based Reasoning Approach for Intelligent Tutoring Systems

ABDELBEEH M. SALEM
Faculty of Computer & Information Sciences
Ain Shams University, Cairo, EGYPT
E-mail: absalem@cis.shams.edu.eg

HISHAM S. KATOUA
Management Information Systems Dept.
Faculty of Economics & Administration
King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia
E-mail: dr_hisham_sa@hotmail.com

Abstract: Case-based reasoning (CBR) is an analogical reasoning method provides both a methodology for problem solving and a cognitive model of people. It is consistent with much that psychologist have observed in the natural problem solving that people do. Intelligent tutoring system (ITS) is a smart knowledge-based tutor and typically have an expert model, student model, instructional module, and intelligent interface. The development of ITSs is a very difficult and complex process that raises a lot of technological and research challenges that have to be addressed in an interdisciplinary way. The new generation of ITS uses the CBR methodology as an efficient paradigm of learning as well as to overcome such challenges. This work discusses the knowledge engineering difficulties and challenges in designing the case-based ITSs. In this context, we suggest in this paper an approach to develop a case-based ITS. The developed prototype has been tested and implemented in biology domain.

Key-words: Machine Learning, Intelligent Tutoring Systems, Case-Based Reasoning, Artificial Intelligence in Education.

1. Introduction

The combination of artificial intelligence (AI) technology and education results in different products of intelligent educational software for all tasks and domains. AI provides a variety of methodologies and theories about reasoning, inference and learning. Hypotheses derived from AI theories can inform curriculum, pedagogy, and potential roles for computers in education. AI based intelligent tutoring systems (ITSs) can adjust its tutorial to the student’s knowledge, experience, strengths, and weaknesses. It may even be able to carry on a natural language dialogue. In addition, automatic generation of exercises and tests is an important feature of these systems [1,2,3].

On the other hand, the field of reasoning is very important for the development of AI-based ITSs [5]. The research area in this field covers a variety of topics, e.g.; automated reasoning, case-based reasoning, commonsense reasoning, fuzzy reasoning, geometric reasoning, nonmonotonic reasoning, model-based reasoning, probabilistic reasoning, causal reasoning, qualitative reasoning, spatial reasoning and temporal reasoning. In fact these methodologies receive increasing attention within the AI in education community.

Rule-based reasoning intelligent tutoring systems (RITSs) solve problems by taking an input specification and then “chaining” together the appropriate set of rules from the rule base to arrive at a solution. Given the same exact problem situation, the system will go through exactly the same amount of work to come up with the solution. In other words RITSs don’t inherently learn. In addition, given a problem that is outside the system’s original scope, the system often can’t render any assistance. Finally, RITSs are very time-consuming to build and maintain because rule extraction from experts is labor-intensive and rules are inherently dependent on other rules, making the addition of new knowledge to the system a complex debugging task.

The idea of case-based reasoning is becoming popular in developing knowledge-based systems because it automates applications that are based
Case-based intelligent tutoring systems (CITSs) may have their greatest impact in complex or “enigmatic” domains (e.g., geometric proofs). ANGLE [10] is a geometric proof tutor which emphasizes student use of schemas in addition to knowledge of rules. Students are encouraged to parse geometric diagrams into meaningful chunks, or Diagram Configuration schemas, which can assist them in planning their proofs. OPERA [16] is an environment in which students can investigate the properties of algebraic operators. Through the use of various schematic representations, students are able to form mental models of concepts such as mappings, functions, and structures of algebraic expressions. Schank’s ASK systems [8], take on the role of expert and guide a user dialog in which the system tells stories to make its points. Archie-2 [6] is used in several architecture studios at Georgia Tech to help student designers with their projects. Its associated authoring tool, Design Muse, is used in classes as well, both to build useful case libraries for several of engineering classes and to give students the opportunity to learn more about some area by preparing and indexing well-articulated cases [7].

In this paper we focus our discussion around the usage of case-based reasoning approach in developing the ITSs. Section 2 gives a brief overview of the general architecture of ITS. The methodology of case based ITS is discussed in section 3. Section 4 presents the knowledge engineering challenges in developing case-based ITSs. Section 5 is dedicated to the proposed model of case based ITS. Section 6 discusses a case study on biology domain. Finally, Section 7 ends with conclusion.

2. The General Architecture of ITS

Different approaches have been developing to integrate machine learning techniques in intelligent learning environments [13,17]. Machine learning is promising for educational application because it leads to the acquisition of new knowledge, the revision and the improvement of the existent knowledge. First, this research inherited the symbolic aspect of the symbolic machine learning. Now, the sub-symbolic techniques (i.e, generic algorithms and neural networks) are being to touch the field of intelligent tutoring systems. The capacity of learning is generally used to cope with the two following problems: (a) student modeling and (b) self-improvement.

Machine learning techniques are useful for ITS, since they touch different research fields such as architecture, interaction, student modeling, knowledge acquisition and problem solving [2,12]. Figure 1 illustrates the main architecture of ITS [5,11]. Student modeling is used to derive an explanation for the student actions. The most important models are: stereotypes overlay models, enumerative theories of bugs, reconstruction of bugs, generation of bugs and combinations of the previous methods. The teaching methods plan the dialogue with the student with didactic background. The user interface is important for the acceptance of the system, so, it should be very good designed to keep the motivation high. Some ITS programs use hypermedia to make them more attractive. Another importance is the use of knowledge base that makes the system able to follow the student in a very flexible way [5]. Most ITS work on procedural, for example mathematical domains, but a lot of work has been done in classification (diagnostics) problem solving. Solving classification problems does not require selection of one or more solutions for a given subset of observations, but also request additional observation that can improve the quality of the solutions.

Fig. 1. The general architecture of ITS
3. The Case-Based Intelligent Tutoring Systems (CITSs).

The general features of CITS are; (a) compose lessons at various levels of knowledge by following the curriculum, (b) solve and generate problems, and (c) generate teaching material. CITS uses an extensive cases of exercises and examples to teach students. The system solves new problems by adapting solutions that were used for previous and similar problems [15].

The methodology of CITS can be summarized in the following two main processes:

(i) **Case-search process.**
In this process the system will search its Case-Memory for an existing similar case(s) that matches the input problem specification (new case). If we are lucky (our luck increases as we add new cases to the system), we will find a case that exactly matches the input problem and goes directly to a solution. If we are not lucky, we will retrieve a case that is similar to our input situation but not entirely appropriate to provide a complete solution.

(ii) **Case-adaptation process.**
In this process the system must find and modify small portions of the retrieved case that do not meet the input specification. The result of case adaptation process is (a) completed solution, and (b) generates a new case that can be automatically added to the system's case-memory for future use.

The technology of CBR directly addresses the following problems found in rule-based approach.
(a) Knowledge Acquisition: The unit of knowledge is the case, not the rule. It is easier to articulate, examine, and evaluate cases than rules.
(b) Performance Experience: A CBR system can remember its own performance, and can modify its behavior to avoid repeating prior mistakes.
(c) Adaptive Solutions: By reasoning from analogy with past cases, a CBR system should be able to construct solutions to novel problems.
(d) Maintaining: Maintaining CBR system is easier than rule-based system since adding new knowledge can be as simple as adding a new case.

4. Knowledge Engineering Challenges
Development of the CITSs faces the following difficulties and challenges;

(i) **Construction of Case-Base/Case Memory.**
In this process we construct the "Case-Base" or "Case-Memory". Each case includes: (a) multimedia description of the problem, (b) description of the correct actions to take including order-independent, optional, and alternative steps; (c) the list of methods to determine whether students correctly executed the steps; and (d) the list of principles that must be learned to take the correct action. This process is very time-consuming to build and maintain because "case" extraction from experts is labor-intensive.

(ii) **Determination of the case features.**
This process is the main knowledge engineering task in developing case-based AI software. This task involves defining the terminology of the domain and gathering representative cases of problem solving by the expert(s).

(iii) **Selection the appropriate authoring tool.**
In this process, the knowledge engineer determines the appropriate intelligent authoring tool/shell which allows the course instructor to enter easily the domain’s knowledge without requiring computer programming skills [14]. It also facilitates the entry of examples/exercises, including problem descriptions, solutions steps, and explanations. The examples may be in the form of scenarios or simulations. It allows organized entry of the course principles and the integration of multi-media courseware (developed with well-known authoring tools) which includes descriptions of the principles or motivational passages. In addition to course
knowledge, the instructor specifies pedagogical knowledge (how best to teach a particular student), and student modeling knowledge (how to assess actions and determine mastery). Some tools were meant for select authors or students and others were designed for a wide set of authors. Some tools were designed to work with a limited area of domain expertise, and some were designed for a wide range of

5. Proposed model for case based Tutoring System

Based on the above discussion, the CITS must have the following features; (a) organizes the knowledge in a lesson-oriented manner. This organization must be dynamically adjusted by the system according to student models, (b) enhances instructor productivity, enabling them to cope with generating more complex training systems required to provide higher skill levels to today's high-tech trainees, and (c) provides tailored instruction and remediation, while allowing flexibility in teaching methods, achieving many of the same benefits as one-on-one instruction.

To achieve the above goals, figure (2) shows our proposed architecture of the CBITS. The system helps students to analyze and repair their solutions. The student inputs a description of the domain situation and his (her) solution and the system can recalls cases with similar solutions and presents their outcomes to me student. Also attempts to analyze the outcomes to provide an accounting of why the proposed type of solution succeeded or failed. The tutoring Case-based module can perform each of the following:
1. Compose lessons at various levels of knowledge by following the curriculum.
2. Solve and generate problems.
3. Generate teaching material.
5. Explain anomalous situation.
6. Generate a Web page.

The web pages module was developed as one of the parts of the system. For each case in the case memory there is a web page. The generated pages are linked according to explicit or implicit relations.

The proposed architecture includes the web pages module and the main concepts of this generator are:
1. For each case in the knowledge base (case-memory/library) there is a web page.
2. The generated pages are linked according to explicit relations.
3. Each object, in Fig. 3, is described as a case in CasePoint environment.
4. The WPGS generates a HTML file for each case and the collection of these files have a set of links among them. These links are automatically written by the system.
5. For each object, a definition may be written by an expert, for example: For the “cetacean” objects; “cetacean, has a backbone, horizontal tail, blow hole for breathing, warm blooded, youngs are born alive, and youngs are nursed with mother's milk”. For the “propoise” concept, derived from the “cetacean” object we can write the following text, “Propoise is approximately 6 feet long, vertical top fin, short blunt snout, lives near the coast”.

Fig. 2. The proposed architecture of CITS
6. Case Study

Our approach was implemented in CBR-Express and CasePoint CBR tools. CasePoint is a run time version of CBR-Express. Developers using CBR-Express don’t interact with it programmatically, but through an excellent if slow interface. The interface deals with all programming elements in case creation or editing. The main features of CasePoint are: (a) fast matching, (b) memory efficient, (c) easily integrated with other applications running under windows, (d) deliver CBR solutions for a wider range of classifications problems at a reasonable cost and (e) automatic case indexing techniques. These features reduce the need to extract and structure specific rule- like knowledge from the experts, the most common consuming parts of traditional ITSs.

The web pages module was developed as one of the part of the case based ITS for teaching sea creatures (cetaceans, fishes, crustaceans and bivalve mollusks). All of the expertise came from a set of encyclopedias and books on sea animals. The system has a case-library/memory, for sea animals structured in the form of cases.

<table>
<thead>
<tr>
<th>Implementation Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long =&gt; 6 feet</td>
</tr>
<tr>
<td>Fine =&gt; vertical top</td>
</tr>
<tr>
<td>Nose =&gt; long beak-like</td>
</tr>
<tr>
<td>Lives =&gt; at sea</td>
</tr>
<tr>
<td>Creature &lt;- cetacean-dolphin</td>
</tr>
</tbody>
</table>

List of respective implementation language:
(long 6-feet) (fin vertical-top) (nose long-beak-like)
(lives at-sea) (creature cetacean-dolphin)

Fig. 3. Sample case for Dolphin on the representation level

7. Conclusion and Future Work

Machine learning techniques are increasing interest in the community working on intelligent tutoring systems. Educational systems integrate machine learning techniques to improve their knowledge to allow a flexibility, versatility and sensibility to learners. More generally, machine learning techniques can be used to achieve the main three objectives: to learn the domain knowledge, to infer the student model and to integrate appropriate pedagogical rules. In this paper, we have described the machine learning technique based on case based reasoning methodology for developing intelligent tutoring system. The proposed technique organizes knowledge in terms of “cases” of past problems and their solutions as well as overcomes the knowledge acquisition difficulty in traditional ITSs. Additionally, the proposed architecture organizes the domain knowledge in a lesson-oriented manner and produces automatic generation of tests and exercises. The developed prototype has been tested and implemented in biology domain. On the other hand, recent advances in Internet technology provide a unique opportunity to distribute training across multiple sites. So, software vendors and researchers start research in order to put the ITSs on the internet for the purpose of distance education through the web [4, 17]. Not only do students receive training at their own sites, but instructor monitor students’ progress from a distance, and course authors maintain and update training material across the Internet.

In our future studies, we are trying to apply our approach in other domains. In addition, future work planned for developing the system includes the distributed artificial intelligence methodology. Through this methodology, the ITS is presented as an open information system in which classical modules, such as the domain expertise module and the tutor, are seen as populations of intelligent actors. Such methodology allows to organize their tasks in a concurrent universe to accomplish a common goal, the teaching session.

References :


