Abstract - Intelligent Tutoring System (ITS) is no more a young research area as it has already revolutionized e-Learning, curriculum instructions and workplace training. For the last few decades, the field has seen approach towards many possible directions and as a result, numerous ITSs have been developed to date. However, they are mostly being studied in research environment and only a few have claimed to be successful in real classrooms for large number of students. This is mainly because human learning phenomena are very complex and itself is an ongoing research activity right through the history of mankind. This paper attempts to identify some key learning/instructional aspects that must be addressed while designing a successful tutoring system. In this regard, we have reviewed some of the well-known ITSs and report an analysis of their success in modeling these instructional ingredients.

Keywords: Intelligent Tutoring System, Student Modelling, Domain Knowledge, Instructional ingredients, Affective factors, Feedback.

1 Introduction

Given a rich intellectual history of decades, Intelligent Tutoring Systems have already attracted attention, funding, and research largely from three different disciplines such as computer science (AI), psychology (cognition) and education and training. Starting from the early days of Computer Aided Instruction (CAI), the field has come across a great success and the recent tutoring systems have claimed to achieve some degree of success in real world applications. However, their success is still limited mainly because it takes a lot of development time (generally 200 hours of development time for one hour of teaching/instructions) and many researchers even describe the ITS development as notoriously costly.

Any software having some degree of intelligent behavior and used for the purpose of learning/instruction is referred to as Intelligent Tutoring system or Intelligent computer Aided Instructions (ICAi). However, human learning is a very complex phenomenon and is being investigated through the lenses of psychologists, psychiatrists, educationist, cognitive scientist and various others. Thus, the success of a tutoring system depends heavily on our ability to understand and model these factors through the four basic ITS modules like domain, pedagogy, student and interface. An ITS may be very much domain rich but adopts a poor policy of providing feedback while another may be good at feedback but pays a little attention towards student’s affective states as well as to well deserved individualized instructions. An ideal tutoring system should be developed keeping in mind all the aspects and thus ITS research must be interdisciplinary in nature. However, most of the existing tutoring systems lack one or more of these aspects. In the next section, we try to enlist the major requirements of a full-fledged ITS, as identified by various experts.

We have studied some of the existing tutoring systems and reported which aspects of learning are being addressed by each of them.

2 Learning Processes

In a real classroom, an expert human teacher who is rich in domain knowledge explains the concepts to the students using various tools and evaluations (mostly through final exam) are conducted to test the mastery of the student over the subject. During these interactions, the teacher tries to model/assess individual student behavior and accordingly decides the pedagogical strategy. A tutoring system should attempt to mimic the best of the collectively human teachers while putting major effort on individualized attention, because this is where teacher fails or falls short to deliver due to time and other constraints.

2.1 Domain knowledge representation

Domain model contains a representation of the information to be taught, provides input into the expert module, and ultimately is used to produce detailed feedback, guide problem selection/generation, and as a
basis for the student model [22]. The domain model may take many forms, depending on the knowledge representation used, the domain it represents, and the level of granularity. A major chunk of ITS development time is supposedly consumed in accomplishing such ingredients. We need to have a standard generic template based domain module which can simply be filled each time by a subject expert and get a quickly developed ITS ready to use. In Cognitive tutors, the domain model consists of low-level production rules that completely describe the expected student behaviour down to atomic thought components while Constraint-based systems describe the possible valid states that an answer may occupy.

2.2 Initial student modeling

Student module plays a very important role in Intelligent Tutoring Systems and stores information about each individual student such as his current state of the domain knowledge, history, and emotional aspects. The content of student models varies widely [27]. Some of these are built for recognizing student plans or solution paths [8], and many others are built for evaluating student performance or problem solving skills [14]; while mostly others are created for describing constraints that the student has violated. We should be clear about what aspects of the student to model while working on a specific tutoring system. Model tracing [15] is the most popular student modeling approach currently. It tracks student’s progress by generating solutions step-by-step, and is suitable for well-defined tasks. But, developing model-tracing tutors for ill-structured tasks is much harder, as it is difficult to come up with problem solvers for such tasks [20]. However, constraint based tutors do not suffer from such difficulties [19].

2.3 Evaluation and Feedback

Simply solving problems through ITS without any feedback may not improve skills or deeper understanding of the subject. Learning occurs best when the learner receives feedback from the system as it improves the learning process on the basis of a continuous assessment of results, the analysis of their quality and performance and feedback for necessary corrections. Feedback encourages desired learning behavior and discourages undesired one, allows understanding how successfully the learner acts, whether he/she applies relevant knowledge and provides opportunities to correct misconceptions. Cognitive tutors try to check if the student’s current solution is on track and, if not, assess what has gone wrong while constraint-based systems evaluate the student solution against the constraints to determine what concepts have been misunderstood.

2.3.1 Simply Hint or Detailed Feedback?

It might be useful to provide a detailed feedback after a session is complete or when a module is finished in order to provide an effective learning experience [4]. Learner would get confidence on the ITS through this as it would lead to a more humanly touch in the learning process. On the other hand, ITS can simply provide hints when error is being made during problem solving. However, the insufficient amount of information in a hint can cause frustration and desire to request the subsequent hints without attempts to solve a problem by the learner.

2.3.2 Immediate vs. Delayed Feedback.

An ITS called E-tutor [21] was evaluated to check the benefit of ‘scaffolding’, ‘hint-on-demand’ and ‘delayed feedback’. E-tutor with dialog led to better learning and represents a more interactive tutor than the “hints on demand” control condition. Also, it was observed that honours students do the best in the delayed feedback condition and regular students do the best in the scaffolding + hints condition. Students who come in with less knowledge benefit more from the scaffolding + hints than students who come in with more knowledge. Students, who come in with more knowledge, benefit from the delayed feedback more than the other groups. With the LISP tutor, Corbett and Anderson [9] showed that immediate feedback leads to three times reductions in the learning time compared to delayed feedback condition. In addition to cognitive benefits, there are also motivational benefits of timely feedback. Students know right away that they are making progress and having success at a challenging task. Further, because the system does not make a big deal out of errors, students do not feel the social stigma associated with making an error in class or on homework.

2.4 Affective factors

Curiosity is considered as an indicator of motivation level and learners with more intrinsic interest display greater levels of pleasure, active involvement, task persistence and lower levels of boredom, anxiety and anger. The role of affective states in learning was investigated from the perspective of a constructivist learning framework through an ITS called AutoTutor [11-13], which teaches introductory computer literacy using natural language based tutorial dialogue. It tracked the learners’ emotions during interactions and these emotions were then correlated with learning outcome measures as shown in table 1. Six different affective states observed are: frustration, boredom, flow, confusion, eureka and neutral [10].

<table>
<thead>
<tr>
<th>Affective states</th>
<th>Learning outcome /Measuring criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frustration</td>
<td>Angry or agitated</td>
</tr>
<tr>
<td>Boredom</td>
<td>Uninterested in the activity or respond slowly to the system</td>
</tr>
<tr>
<td>Flow</td>
<td>Shows interest, pays attention and respond quickly</td>
</tr>
<tr>
<td>Confusion</td>
<td>Puzzled, not sure how to continue or struggling to understand the material</td>
</tr>
</tbody>
</table>
can be applied for a different domain. It works on the domain independence. Domain expert only need to edit separation of processing from data and this allows another feature of the current architecture is its modules without redesigning the system from scratch. The major advantage of this architecture is the openness which means new module can be easily accommodated or major changes can be made to existing modules without redesigning the system from scratch. Another feature of the current architecture is its domain independence. Domain expert only need to edit the knowledge base stored in external files and the system can be applied for a different domain. It works on the cognitive models of self-regulated learning, which says that the student should regulate key cognitive and metacognitive processes in order to learn about complex and challenging science topics. The system tries to classify three different student mental models (low, medium and high) by combining his prior knowledge activation data with various supervised machine learning algorithms. MetaTutor does not have any feedback mechanism yet though the system is still in its various implementation phase.

### 3.1 MetaTutor

MetaTutor [23] is an intelligent tutoring system to teach students how to generate smaller subgoals of a problem while learning science topics and to generate necessary feedback. It is a complex system and consists of nine major logical components: pre-planning, planning, student model, multi-modal interface, feedback, scaffolding, assessment, authoring, and system manager. The major advantage of this architecture is the openness which means new module can be easily accommodated or major changes can be made to existing modules without redesigning the system from scratch. Another feature of the current architecture is its domain independence. Domain expert only need to edit the knowledge base stored in external files and the system can be applied for a different domain. It works on the cognitive models of self-regulated learning, which says that the student should regulate key cognitive and metacognitive processes in order to learn about complex and challenging science topics. The system tries to classify three different student mental models (low, medium and high) by combining his prior knowledge activation data with various supervised machine learning algorithms. MetaTutor does not have any feedback mechanism yet though the system is still in its various implementation phase.

### 3.2 AnimalWatch

AnimalWatch [5, 7], an ITS designed for students mastering basic computation and fractions skills. It uses problem solving errors to estimate the student’s skill with each math topic, and selects problems that the student should be able to solve by using the integrated help resources. When the student can solve challenging problems in one topic successfully, the system will move on to a new math topic. One encouraging approach in this study is the pre and post test based evaluation methodology. This can be implemented in future ITS systems where students can be asked to go through a pre-test for each topic and the result can be used by the student module to decide the teaching strategy for this student. Recently, the system has got immense use by the online users and the webpage claims that more than 350,000 AnimalWatch word problems have been completed since August 2010.

### 3.3 SQL Tutor / SQLT-Web

Mitrovic et al. [16, 17] developed SQL Tutor, an ITS for SQL database language. It contains no module and uses Constraint-Based Modeling to model knowledge of students. At the beginning of a session, SQL-Tutor selects a problem for the student to work on. When the student enters a solution, the pedagogical module sends it to the student modeller, which analyzes the solution, identifies mistakes (if there are any) and updates the student model appropriately. On the basis of the student model, the pedagogical module generates an appropriate pedagogical action (i.e. feedback).

SQLT-Web [18] is a Web-enabled version of the SQL-Tutor and was developed reusing the components of the standalone system. SQLT-Web maintains a centralized repository of student models and supports multiple simultaneous students, thus giving students freedom to access the system at any time and from any place. The session manager records all student actions and the corresponding feedback in a log. It also requires the student modeller to retrieve the model for the student, if there is one, or to create a new model for a student who interacts with the system for the first time.

### 3.4 ELM-ART tutor

ELM-ART [6, 25] is an intelligent interactive educational system to support learning programming in LISP. It provides all learning material online in the form of an adaptive interactive textbook. Using a combination of an overlay model and an episodic student model, ELM-ART provides adaptive navigation support, course sequencing, individualized diagnosis of student solutions, and example-based problem-solving support. ELM-ART is based on ELM-PE [26], an Intelligent Learning Environment that support example-based programming, intelligent analysis of problem solutions, and advanced testing and debugging facilities. The negative side of this method is that there is a higher risk for the student to get lost in this complex hyperspace.

### 3.5 Andes Physics Tutoring System

Andes [24] is an intelligent tutoring system that helps university students to learn physics. It concentrates on web-based homework (WBH) and gives immediate feedback. Andes provides three kinds of help during learning:

- Andes pops up an error messages whenever the error is probably due to lack of attention rather than lack of knowledge. For example, leaving a blank entry in a dialogue box, using an undefined variable in an equation, or leaving off the units of a dimensional number.
- Students can request help through a help button to ask “what’s wrong with that?”.
If students are not sure what to do next, they can click on a button that will give them a hint. This is called Next Step Help.

Multiple evaluations have reported Andes to be significantly more effective than doing pencil and paper homework.

### 3.6 PAT

PUMP Algebra Tutor (PAT) [15] was developed by the Pittsburgh Urban Mathematics Project (PUMP) where students engage in investigations of real world problem situations and use modern algebraic tools to solve problems and to communicate results. It is based on ACT theory [3] and cognitive tutoring technology [1, 2]. The cognitive model is written as a system of if-then production rules that are capable of generating the multitude of solution steps and mis-steps typical of students. During student’s interaction with the PAT, the tutor monitors their activities, and provides feedback on what they are doing. For the most part, the tutor is silently tracing student actions in the background. When a student makes an error, it is "flagged" without comment, which appears to reduce students' negative feelings associated with making errors in math class. But if the student's error is a commonly occurring slip or misconception that has been codified in a buggy production rule, a message is provided that indicates what is wrong with the answer or suggests a better alternative.

### 4 Analysis

Table 2: ITS and characteristics

<table>
<thead>
<tr>
<th>Properties</th>
<th>Domain Covered</th>
<th>Feedback</th>
<th>Use of Animated Agent</th>
<th>Student model</th>
<th>Other notable features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta Tutor</td>
<td>Human body</td>
<td>verbal</td>
<td>Yes</td>
<td>Adaptive Hypermedia</td>
<td>Self regulated learning</td>
</tr>
<tr>
<td>Animal Watch</td>
<td>Math Teaching</td>
<td>immediate</td>
<td>Yes</td>
<td>Hidden Markov Models</td>
<td></td>
</tr>
<tr>
<td>SQL Tutor</td>
<td>Learning SQL Query</td>
<td>5 levels of feedback</td>
<td>No</td>
<td>CBM for short-term and Overlay for long-term modelling</td>
<td></td>
</tr>
<tr>
<td>ELM ART</td>
<td>LISP Programming</td>
<td>Immediate</td>
<td>No</td>
<td>Episodic/Collaborative (open, editable)</td>
<td>Display material as adaptive interactive book</td>
</tr>
<tr>
<td>ANDES</td>
<td>Physics</td>
<td>Immediate</td>
<td>No (instruction through videos)</td>
<td>Bayesian Network</td>
<td></td>
</tr>
<tr>
<td>PAT</td>
<td>School level Algebra learning</td>
<td>Immediate (timely)</td>
<td>No</td>
<td>model tracing and knowledge tracing (ACT theory)</td>
<td></td>
</tr>
</tbody>
</table>

LISP tutor [9] shows that immediate feedback is more beneficial compared to delayed feedback. AutoTutor [10] concludes that emotional states like confusion and flow are positively correlated to learning while boredom is negatively related. The correlations between learning and states like eureka and frustration is low and least significant. E-tutor [21] tells that students having less domain knowledge benefit more from the scaffolding + hints than students with more knowledge and students who having more knowledge benefit from the delayed feedback more than the other groups. The success of MetaTutor [23] raises hopes of achieving domain independence as its architecture separated the domain module from the rest of the parts. AnimalWatch [5, 7] employs the pre and post test based evaluation method for initial student modeling and its update.

Table 2 summarizes a comparative view of the six ITSs we studies in section 3. Our evaluation here is based on major parameters like: domain they cover, feedback policy being employed, pedagogical agent (if any) and the student modelling methodology being followed.

### 5 Conclusion

Throughout the paper, we have tried to describe the important design specifications for a generic intelligent tutoring system through our experience of evaluating some of the prominent ITSs and their results. Various learning/instructional ingredients are being identified and their effectiveness is explained. For example, the benefit of immediate vs. delayed feedback, small hints vs. detailed help etc. are compared. ITS development is a costly affair in terms of cost and complexity. We need to develop standardized reusable components which can be used to develop any domain specific tutor in a short span of time and in a cost effective manner. In future works, we plan to model a general purpose tutoring architecture emphasizing the role of emotional factors during a learner’s interaction with the ITS.

### 6 References


