Chapter 2 : LITERATURE SURVEY

Section 2.1: Literature survey of Intelligent Tutoring Systems (ITS):

The literature survey is focussed on student models (SM) and application of machine learning techniques to SM generation.

Section 2.1.1: Literature survey of Student Models:

Definition:

1) The SM is that part of an ITS that enables it to answer questions about the student using the system [Self 1988]. The questions are of the form: what can the student do, what does the student know, what type of student is she and what has the student done.

2) The component of an ITS that represents the student’s current state of knowledge is called the SM [VanLehn 1988a].

Student models have six possible uses [Self 1988]. They are:

1) Corrective functions: To help eliminate errors in the student’s knowledge. The methods are bug identification,
remediation, indirect remediation, counter example, solution tracing, retrospection and tactical withdrawal.

2) **Elaborative functions**: To help complete the student's knowledge. The choice of the next topic is done by (a) curriculum driven, (b) expert-student comparison, (c) internal analysis of the student's knowledge and (d) learner (student) controlled.

3) **Strategic functions**: To help initiate changes in the tutoring plan; example: in the LISP ITS [Anderson & Reiser 1985] if the tutor encounters two consecutive bugs that it cannot analyse, the design mode is invoked. In this mode the tutor tries to help the student design the algorithm. The system then reverts to the coding mode.

4) **Diagnostic functions**: To help diagnose student errors. One approach to locate the errors is to choose a problem that is more discriminating in highlighting the bug. A second approach is to interact with the student in detail, presenting discriminating examples.

5) **Predictive functions**: To help determine the student's likely response to the tutorial action.

   a) **Performance prediction**: Predicting the student's performance is of use only when the performance differs from the
prediction. NEOMYCIN embarks on a data driven analysis of the performance at such points of difference [Clancey & Letsinger 1984].

b) Learning prediction: If a model of the student's learning process is available it can be used to predict the effect of didactic actions. The system would then have a basis for computing the new SM resulting from the didactic action. Also the system would give an analytical reason for selecting that action, by running the learning procedure with a set of potential actions and selecting that which is predicted to have the most effect (example, CTP [Self 1977]).

6) Evaluative functions:
(a) Student evaluation: A description of what the student is thought to know can be made.
(b) System evaluation: Utilising a simulated student, a measure of the system's potential effectiveness can be got.

Types of student models:
1) Performance measure: This is computed considering the student's answers in examinations. As the knowledge that a student has acquired is not reflected, performance measures are not useful in deciding how to teach a student a particular concept [Anjaneyulu 1989].

26
2) **Differential Model**: In this modelling technique the student's behaviour is analyzed and recorded in relation to the expert's choice for the same circumstances. In the WEST project differential modelling was used to find the relevant issues to be taught and the corresponding appropriate examples [Burton & Brown 1976]. This modelling technique has also been employed by ENIGMA, an ITS centered on a cryptic arithmetic problem [Forcheri & Moltino 1991].

3) **Overlay models**: This is similar to differential models. It also attaches individual evaluations to single units of knowledge by comparing the student's behaviour with the expert's. The difference is that as these units are parts of the expert itself the knowledge of the student is viewed as a subset of the expert's knowledge [Carr & Goldstein 1977].

4) **Genetic graph**: This combines the concept of overlay on a rule based representation with a learner oriented set of links that represent the relationship between the rules [Goldstein 1978]. The word genetic which refers to Piaget's notion of genetic epistemology is used as the system tries to capture the evolutionary nature of knowledge [Wegner 1987].

5) **Simulation models**: Also called as executable or runnable models these are useful in yielding exact predictions about the behaviour of a particular student in a particular context. This
allows verification of hypotheses (example, STEAMER [Hollan et al. 1984]).

6) Buggy models: This attempts to represent a student’s knowledge in terms of a catalogue of bugs or misconceptions. Bugs are usually procedural or localised errors rather than deep, pervasive misconceptions (example, Buggy [Brown & Burton 1978], Debuggy [Burton 1982]). A bug catalogue is also used in SAMPLE [Micarelli et al. 1992].

7) Student modelling using fuzzy values: The student model is represented as a set of variables. Each student modelling variable is represented as a fuzzy variable. In SHERLOCK II [Lesgold et al. 1990] a fuzzy variable is considered as a distribution over the set of possible knowledge states of a student. Each student modelling variable is given five such knowledge states: no knowledge, limited knowledge, unautomated knowledge, partially automated knowledge and fully developed knowledge.

8) Mental models: This is a coherent collection of knowledge held by a person about some aspect, entity or concept of the world. As such, it forms a deep conceptual model [Bhuiyan et al. 1990]. This knowledge can help in predicting the next behaviour of the student. A notion of components of mental
models considers a global mental model of a student as a combination of different component models for each subprocess [Collins & Gentner 1983 and Collins 1985]. This leads to the concept of evolution of mental models [Wegner 1987].

Simulation models can be advantageously used to demonstrate the problem solving process on each problem given to the student. Unfortunately they are very difficult to derive. In the differential, overlay and genetic graph models, variations in the student's knowledge are assumed to result from incompleteness but never from incorrectness. Another evidentially supported viewpoint, considers incorrect versions of knowledge entities or even incorrect general beliefs as the primary cause for manifested errors. This viewpoint is supported by buggy models. Though buggy models do not represent factual errors or deep misconceptions they do list internalised procedural errors [Wegner 1987].

The theory of bugs has been the primary focus of interest, in recent times. Three fundamental types of theories of bugs are present [Wegner 1987]: enumerative, reconstructive and generative.

Enumerative: The system is supplied with catalogues or libraries of commonly observed errors or misconceptions. They
are usually represented as parts of a process model. This allows runnable models incorporating errors. This can also be used to 'parse' the student's actions.

If the bugs are represented as primitives that can be combined then they can be used to provide explanations of observed errors. The combination of the primitives results in combinatorially large search spaces (example, LMS-II [Sleeman 1983]). Also the problem of multiple bugs, is not addressed.

**Reconstructive**: These systems attempt to reconstruct bugs from observed errors in a data-driven bottom-up fashion (example, ACM [Langley et al. 1984]). PIXIE's [Sleeman 1987] approach is to hypothesise new mal-rules (incorrect variants of correct rules) using the student's steps whenever these steps cannot be recognised by the existing set of rules.

**Generative**: The attempt here is to reduce observable errors to some underlying misconceptions or to the mislearning of some aspect of knowledge. REPAIR [Brown & VanLehn 1980] - STEP [VanLehn 1987 & VanLehn 1988b] theory provides an independent framework for explaining the origin of bugs. The theory claims that student's reach an 'impasse' during problem solving. An impasse occurs when the step that they believe should be executed next cannot be performed. At this juncture
the student will attempt a 'repair'. Repair is an experimental series of operators that the student will attempt on the procedure in order to proceed. The repairs attempted were presented as dependent, only on the procedure and its current impasse and independent of how the incomplete procedure was derived. Matz [Matz 1982] views bugs as a result of reasonable though unsuccessful attempts to adapt previously acquired knowledge to new situations. [Escott 1988] confirms use of analogical reasoning by a student in the course of learning to program in LISP. She links errors in student's programs to misuse of analogy.

Though a dynamic generative theory of bugs has not been incorporated into a complete tutoring system, such theories still, provide fresh avenues for remediation.

Section 2.1.2 : Machine learning application to student models:
Leeds Modeling System-II (LMS-II) [Sleeman 1983] utilised a library of mal-rules. LMS-II used the learning strategy of induction (constructive induction) to infer the student's model, by observing the student's response to a set of problems. The responses formed a set of examples. The system was unable to handle situations of multiple bugs and unanticipated or inconsistent bugs.
Automated Cognitive Modeler (ACM) [Langley et al. 1984] employs the techniques of exhaustive search and discrimination learning to construct procedures that account for student's answers. ACM was applied to the subject domain of multi-digit-subtraction. The exhaustive search technique limits the application to a narrow range of school subjects, wherein, the problem space is small enough. From the psychological point of view, ACM has three shortcomings: the absence of goal structures, the poor modelling of attention processes and the non-addressing of short term memory limitations [Langley et al. 1984]. Another system employs the technique of explanation based learning for student modelling [Bar-on & Bachor, 1988]. Two different paths are followed: (1) Learning the student's wrong conception and (2) learning the misconception, leading to the mistake. Generalisation of the explanation in the first part, leads towards a generalised, conceptual graph of the student's wrong conception, while the second path leads to a generalised misconception [Bar-on & Bachor 1988].

Adaptive Control of Thought (ACT) and its successor ACT* [Anderson 1983] were designed to embody a general theory of cognition with an emphasis on skill acquisition. The goal restricted production system architecture of the theory was implemented as a computer model (GRAPES). This served as a
learning model for tutoring systems built around ACT. The basic learning strategy is knowledge compilation. Knowledge compilation can be done by proceduralisation of a general piece of knowledge and by composition of a few rules, normally used in a sequence, into a single rule. This work gave rise to the LISP tutor (GREATERP) and the Geometry tutor [Anderson & Skwarecki 1986].

SIERRA [VanLehn 1987] is a program that learns incrementally from examples by completing explanations. The system demonstrated unbiased inductive learning using the constraint that only one new subprocedure (or concept) (or disjunct) was introduced in a lesson. This solved the problem of disjunction that generalisation learning strategies face. To address the problem of invisible objects that inductive learning faces, the show-work condition was introduced. This condition requires all intermediate values generated in the course of a computation or problem solving to be stated explicitly. As an extension of work on impasse and REPAIR theories, VanLehn proposed the STEP theory based on work with SIERRA. Three main sources of impasses were noted: 1) mistaken or excessive generalisation 2) overspecificity and 3) deletion of rules. STEP theory was unable to explain the problem posed by
differences within acquired critics (critics that work on repair heuristics), as explaining this would require a theory that could account for individual differences in the acquisition of general guidelines for the domain [Wegner 1987].

While ACM was purely diagnostic with no attempt made to model the learning approach of the student, SIERRA used the learning technique of induction to model the way students actually learn their procedures from a set of correct examples in an instructional sequence.

Section 2.1.3: Machine learning based, generation of the other models of an ITS:

Application of machine learning techniques to the other models of an ITS, has been limited. O'Shea [O'Shea 1982] and Kimball [Kimball 1982], present self improving tutor models in the areas of quadratic equations and symbolic integration, respectively. These aim to achieve five characteristics:

1. Transmit, problem solving heuristics.
2. Generate appropriate expository examples.
3. Deal with arbitrary student initiated, examples.
4. Handle a wide range of student backgrounds.
5. Acquire superior problem solving approaches from the students themselves.

Kimball, employed primarily two strategies:
(a) The tutor provides approach choice advice, by presenting the student with its own approach choice priorities.

(b) A 'better' solution by the student is used in further passes by the tutor. 'Better' was defined as a solution path, shorter than one the tutor had [Kimball 1982].

O'Shea, employed a deductive procedure to make experimental changes on the production rules, experiencing the teaching strategy. A statistical evaluation followed, resulting in an update [O'Shea 1982]. This primarily, exemplified the learning by experimentation strategy. The major difficulty is in deciding what, experimental changes are to be made in each run. The limitations of this technique, include a vulnerability to problems of local minima and to a variant of the frame problem [O'Shea 1982].
Section 2.2 Literature survey of Machine learning systems:

The literature survey in this section is focussed on integrated machine learning systems.

Section 2.2.1 Literature survey of integrated machine learning systems:

The integration and interplay between the learning strategies is a major cause for the excellent, human learning capabilities. The idea of an integrated machine learning system was first mentioned by Michalski and Lenat in a panel discussion [Michalski et.al. 1986].

An integrated machine learning system is built with the primary motive of overcoming the limitations of individual learning strategies. This research line has led to two approaches:

(a) Cascaded multi-strategy learning systems: The system utilises a cascade of more than one learning strategy, where the output of one learning strategy forms the input of another learning strategy, with a preset control. For example: [Lebowitz 1986], [Flann & Dietterich 1989], [Berzadano & Giordana 1988], [Mooney & Ourston 1989], [Star 1989], PRODIGY [Minton et.al. 1989], [Ahmad et.al. 1991], [Hunter 1993]. [Lebowitz 1986] and [Star 1989] use similarity based learning before
explanation based learning, while [Berzadano & Giordana 1988] and
[Mooney & Ourston 1989] use explanation based learning before
similarity based learning. The programs of [Lebowitz 1986],
[Berzadano & Giordana 1988] and [Mooney & Ourston 1989] were all
used in instances of EBL systems with incomplete theories of the
domain. [Hall 1988] used analytical procedures when EBL found the
subject theory to be inadequate. [Duval 1991] uses inferences
made by abduction and analogy on the partial explanations of an
EBL system. PRODIGY was first developed with the EBL learning
strategy. The other learning modules in the design were:
Learning by experimentation and learning by analogy
(derivational analogy) [Minton et.al. 1989].

[Veloso & Carbonell 1991] present an evaluation of
analogical reasoning within the PRODIGY framework.

(b) Non cascaded multi strategy learning systems: These
systems apply in a predefined order, a next learning strategy if
the first fails, and so forth. For example, OCCAM [Pazzani 1988],
[Genset et.al. 1990].

Sarrett and Pazzani [Sarrett & Pazzani 1989] presented
a framework where separate hypotheses are formed using
explanation based learning and similarity based learning methods
and these are then combined to form a composite hypothesis.
All the above systems rely on one fixed sequence of invocation of learning strategies. The problem with this approach to integrated machine learning is that: as different problem domains require different sequences of application of learning strategies, this limits the generality of such systems, confining them to the problem domain for which they have been developed. A more sophisticated scheme employs different hand-coded sequences of learning strategies applicable to different inputs depending on the situation. A proposal for such an integrated machine learning system was put forward by Tecuci [Tecuci 1991]. The system proposed an integration of EBL, learning by analogy, SBL, learning by asking questions and by being told, abduction and conceptual clustering [Tecuci 1991]. In this paradigm computationally efficient selection of the learning algorithm is important [Cox & Ram 1992].

Meta-AQUA [Cox & Ram 1992], is a program that performs multistrategy learning through self-analysis of its reasoning process. Meta-AQUA is the first attempt at an integrated machine learning system capable of selecting the appropriate learning strategy for a situation by itself, without the help of a hand-coded sequence of learning strategies. Meta-AQUA uncovers the problem of learning strategy selection. The
Classes of learning situations are identified based on an analysis of the types of reasoning failures that occur. Each type of reasoning failure is characterised by a description of how the conclusions were drawn, why the conclusions were faulty, what the correct conclusions ought to have been and how the reasoner ought to have drawn them. The system analyses the situation to determine the type of reasoning failure using Meta-Explanation structures. Each type of reasoning failure is associated with what needs to be learned and learning strategies that can learn the desired knowledge. This knowledge was represented using Meta-explanation structures [Cox & Ram 1992].

Section 2.3 : Comments on literature survey :

Section 2.3.1 : Comments on the literature survey of Intelligent tutoring systems :

The literature survey indicates the trend towards the representation of student models by a theory of bugs. This has been so as it is very important to perceive the student's knowledge exactly. REPAIR-STEP [VanLehn 1987], [Escott 1988] and [Matz 1982] have been attempts at producing an explanatory theory of occurrence of bugs. While [VanLehn 1987] focuses on wrong inductive learning as the cause for errors, [Matz 1982]
and [ Escott 1988 ] focus on misapplied, learning by analogy as the cause. Many of the student modelling techniques, especially the buggy models have been developed in the subject area of multi-digit-subtraction (example, Buggy [Brown & Burton 1978], [Brown & VanLehn 1980], Debuggy [Burton 1982], [VanLehn 1987]).

The application of machine learning techniques to student model generation has been noted to be restricted. SIERRA [VanLehn 1987], ACM [Langley et al. 1984], CTP [Self 1977], ACT [Anderson 1983] are some of the attempts. Section 2.3.2: Comments on literature survey of machine learning systems:

The focus on integrated machine learning systems highlights the two approaches of Cascaded and Non-Cascaded, prior to meta-AQUA [Cox & Ram 1992]. In addition to the problem in the above two approaches mentioned within the literature survey, an additional problematic requirement for these approaches is:

A complete set of all possible combinations, of sequences of learning strategies, with their corresponding application domains is required. Also, completeness of the algorithms that choose the right sequence of learning strategies is important.

The approach of meta-AQUA to the learning strategy selection problem was to use the taxonomy of reasoning
failures linked to the appropriate learning strategy. Linking a reasoning failure to a learning strategy assumes that a particular type of reasoning failure would always require the same learning strategy irrespective of the environment. [Ram & Cox 1993] give three types of reasoning failures in meta-AQUA: novel situations, incorrect background knowledge and mis-indexed knowledge structure. Under 'novel situations' the learning strategies applicable are mentioned as: Explanation Based Generalisation, inductive generalisation from examples and Explanation Based Refinement coupled with index learning. They have not been able to confine the link to only one learning strategy. (They have also not mentioned the strategy of learning by being told, which is definitely applicable.)