Agent Based Student Modeling in Distributed CBR based Intelligent Tutoring System

O.P.Rishi[1], Rekha Govil[2] and Madhavi Sinha[3]

Abstract— **Student modeling is a key component in any Intelligent Tutoring System (ITS). In today's distributed computing environment, the tutoring system can take advantage of networking to utilize the model for a student for students from other similar group. In the present paper we present a methodology where using Case Based Reasoning (CBR), the ITS provides student modeling for online learning in a distributed environment with the help of agents.**

The paper describes the approach, the architecture, and the agent characteristics for the student modeling in the ITS. This concept can be deployed to develop ITS where the tutor can author and the students can learn locally whereas the ITS can model the students' learning globally in a distributed environment. The advantage of such an approach is that both the learning material (domain knowledge) and students' model can be globally distributed thus enhancing the efficiency of ITS with reducing the bandwidth requirement and complexity of the system.

***Index Terms*— Student Modeling, Intelligent Tutoring System (ITS), Distributed System, Intelligent Agent, CBR.**

I. Introduction

The major challenge in teaching is to improve both instructional productivity and learning quality for large and diverse population of students under real world constraints such as limited financial resources and insufficient number of qualified instructors [1]. The different researches suggest that students who are engaged in learning through intelligent tutoring processes are more likely to achieve success [2].

Since mid 90's, few educational system models were web based and used ITS with student modeling in distributed style. In past decade, researchers from different disciplines have come out with systems which define and classify different teaching and learning styles in distributed environment.

In this paper we conceptualize a Case Based Distributed Student Modeling (agent based) ITS architecture to support student-centered, self-paced, and highly interactive learning. In this system the first step in building an effective learning environment is building a Case Base where the system maintains a rich set of cases (scenario) of student learning pattern, and employs an efficient and flexible case retrieval system during a session between the student and ITS. This maximizes the interactivity between the ITS and the students; and customizes the learning process to the needs of an individual student [3]. The system must use the student's learning profile such as learning style and background knowledge in selecting, organizing, and presenting the learning material to support case based learning. It also supports personalized and more intensive interaction between the student and the ITS [4]. Distributed CBR based student modeling enables adaptive delivery of educational content and facilitates automatic evaluation of learning outcomes. The system also incorporates a new approach to course content organization and delivery, which can be developed based on distributed and agent based instructional components. Instructional components represent the customized interactive presentation of any topic of the subject or different subjects. They must be self-contained, autonomous, and easily integrable into a wide range of courses.

II. Proposed System for Distributed Case Based Reasoning

The proposed system is based on finding a case that is similar to the learning domain of a past student in a distributed environment. A similar learning domain means that when a student enrolls in the system and on presentation of content, questions, examples, and assignments the student responds. CBR Based student modeling system compares the student's solving process with the patterns of responses stored in the case base. At this time, the inference engine reasons the state of the student's knowledge through the student's solving process. If a new situation occurs then case based system treats this as a new case and records it, termed as machine leaning.

The proposed distributed CBR based student modeling system does its work with the help of a number of specialized agents with different expertise. There is a *personal agent* that manages the student's personal profile, including knowledge background, learning style, interests, courses enrolled in, etc. The *personal agent* talks to other two agents in the system namely t*eaching agent* and c*ourse agent* through different

O. P. Rishi is with the Birla Institute of Technology, Extension Centre Jaipur, BISR campus, 27 Malviya Industrial Area Jaipur 302017 INDIA phone: + 91 94142 58030; fax: + 91 141 2751601; (e-mail: omprakashrishi@ yahoo.com / oprishi@bitmesra.ac.in).

Rekha Govil is with the Apaji Institute of Mathematics & Applied Computer Technology, Banasthali Vidyapith, Rajasthan-304022 INDIA (e-mail: grekha@banasthali.ac.in,).

Madhavi Sinha is with the Apaji Institute of Mathematics & Applied Computer Technology, Banasthali Vidyapith, Rajasthan-304022 INDIA (e-mail: smadhavi @jp.banasthali.ac.in).

communication channels situated in distributed environment (Figure-1).

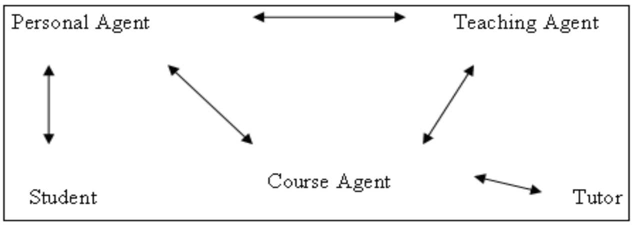

Figure: 1. Communication model among Agents

A web based course is supported jointly by a *teaching agent* and a *course agent,* that manage course material and course-specific teaching techniques and student modeling strategy. Multiple *course agents* may exist on distributed sites to provide greater efficiency, flexibility, and availability. The *teaching agents* can talk to any *course agent*, and often choose one nearby for better performance. The *course agents* also act as mediators for communication among students and tutors. A *teaching agent* interacts with a student and serves as an intelligent tutor of a topic or course. From a *course agent*, each *teaching agent* obtains course material and course-specific teaching techniques and then tries to teach the material in the most appropriate form and pace based on the background and learning style of the student [5]. Lecture note, presentations, multiple examples with different difficulty level are used to make difficult concepts and operations easy to understand.

III. Architectural Design of CBR based Student Modeling in Distributed Environment

CBR Based student modeling system can be divided into three parts. A Case Based ITS (Intelligent Tutoring System), a student interface module, and, the instruction processing engine or agent communication system. The ITS is divided into a student module, a tutor module with expert module (human tutor interaction), and an interface module. The student module consists of a Case Base, which is based on an overlay or bug part library system [2].

In the proposed distributed environment student module, tutor module and case based student modeling reside on every node of the network. *The teaching agent* indexes all the cases globally and *personal agent* is responsible for indexing all the registered or aspirants students (learner) with their personal profiles like static background profile (such as GPA or percentage, majors, interests and course taken) and the students' dynamic profile globally (based on their interaction on the agent) [6] . The *course agent* manages the course material as per students requirement and *teaching agent* decides the topics or lessons, to be supplied to the students according to student performance. The student modeling keeps the record in case base for future reference and manages the student performances and course material accordingly. The Network and nodal architecture of the proposed system are shown in figures 2 (a) & (b) respectively.

Thus the whole system is managed in the distributed environment with just three agents:

1. Personal agent,

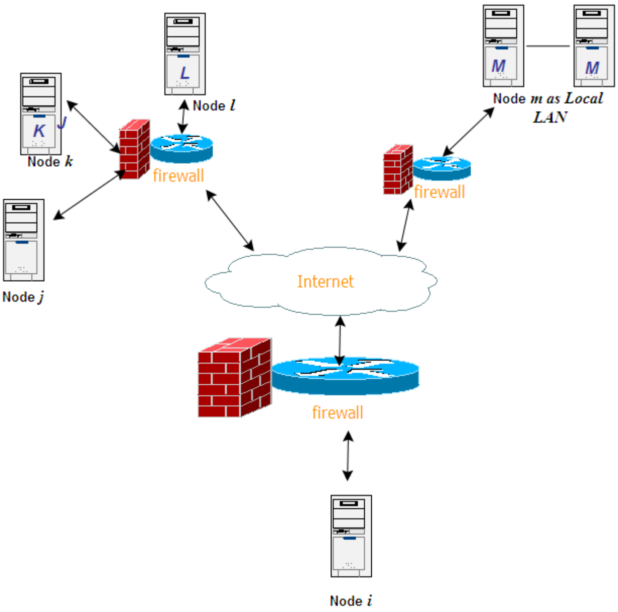

Fig 2 (a) Network architecture of CBR based ITS in Distributed & agent based environment

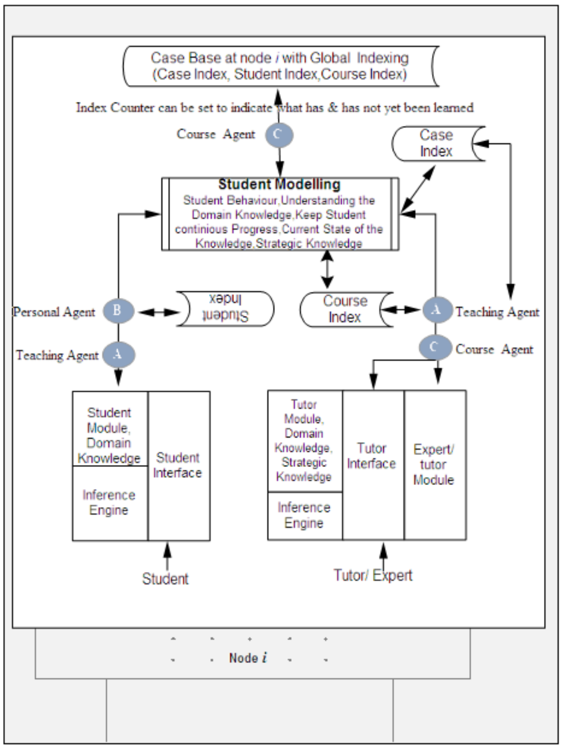
Case Base at node *i* with Global Indexing
(Case Index, Student Index,Course Index)
Index Counter can be set to indicate what has & has not yet been learned
Course Agent
C
Case Index
Student Modelling
Student Behaviour,Understanding the Domain Knowledge,Keep Student continious Progress,Current State of the Knowledge,Strategic Knowledge
Personal Agent
B
Student Index
Course Index
A
Teaching Agent
Teaching Agent
A
C
Course Agent
Student Module, Domain Knowledge
Inference Engine
Student Interface
Tutor Module, Domain Knowledge, Strategic Knowledge
Inference Engine
Tutor Interface
Expert/ tutor Module
Student
Tutor/ Expert
Node *i*

Fig 2 (b) Nodal architecture of CBR based ITS in Distributed & agent based environment.

2. Teaching Agent, and
3. Course agent

In the proposed learning system there are also three global indices: Student Index, Case Index, and Course Index. The Student Index maintains the record of system log file (IP addresses of the entire network) and student log file (User ID) registered in the entire network managed by the personal profiler or *personal agent*.

The *personal agent* maintains the students' profile master, students' learning profile, and usage profile. In profile master user ID, registered node, user personal profile, course ID, and preliminary or past performance, level of student (poor, average, excellent), category of the student(slow learner, normal learner, fast learner) are stored. Similarly, learning profile maintains the record of subject ID, topic ID and topic name, pre-leaning performance, and post learning performance, performance of individual topic and subject, type of errors performed by the student. The usage profile keeps the record of system ID, time in, time out, topic name, number of attempts, questionnaire etc.

The Case Index maintains the record of cases managed by the *teaching agent*. In Case Index, case type, content of each case, case adaptation condition, case indexing strategy, case retrieval method, case recall method, case updation (addition, deletion, modification) methods, error rate, teaching strategy, location of the case where it is stored, and the remedial strategy used by the system, are stored and maintained.

The *teaching agent* decides the teaching strategy by using learning profile of the student and thereafter presents the right content to the student with the help of *course agent* using Case Index. The Course Index managed by the *course agent* maintains the record of available course content like ordering or sequencing of course material, set of related examples, assignments, questionnaires, a set of exercise problems to access the student's understanding of the topic and remedial content.

For the purpose of fault tolerance all storage in the distributed environment are maintained in duplicate. It means if a new case is diagnosed and stored at node *i*, a copy of it is also stored on node *j*, *j* being the next nearest node available. Later, one can research on optimum redundancy parameters. Thus with the help of these three agents; personal agent, teaching agent and course agent, and three global indices; student index, case index and course index the complete learning process is managed.

Let us think of a scenario of a student interacting with the ITS:

1) The student logs in either as a new user or a registered user from any node *i* of the network. In the first case the *personal agent* will record the student's personal profile at node *i* and update the global student index. In the second case there could be two possibilities : node *i* is the node which contains this student's profile, or , the student's profile is present at any other node *j* of the network (*j=1, n; j<>i*). In any of these two cases the Student's personal profile shall be available to the system with the help of the *personal agent*.
2) Student starts session with the ITS. If new user, then the ITS first assesses the student's level and creates his/her learning profile whereas for a registered user it obtains this information with the help of *personal agent*. The *teaching agent* then decides the pedagogy and proceeds with the session. It obtains the desired courseware with the help of *course agent*.
3) The learning behavior of the student is continuously, monitored by the *teaching agent,* and compared with the previous case, stored in the Case Base. If a new pattern is observed the *teaching agent* records it as new case and flashes the message to the *Course Agent* for obtaining the right courseware for the case. This is done through Tutor Module of ITS which is not dealt with in the present paper. Thus the system employs Machine Learning.
4) Student goes through the learning content with in finite time and goes through the self assessment session. The *Personal Agent* records the performance (learning extent, learning rate and pattern) and takes further action based on Student Modeling output which uses CBR. The complete flow of the process is shown in figure 3.

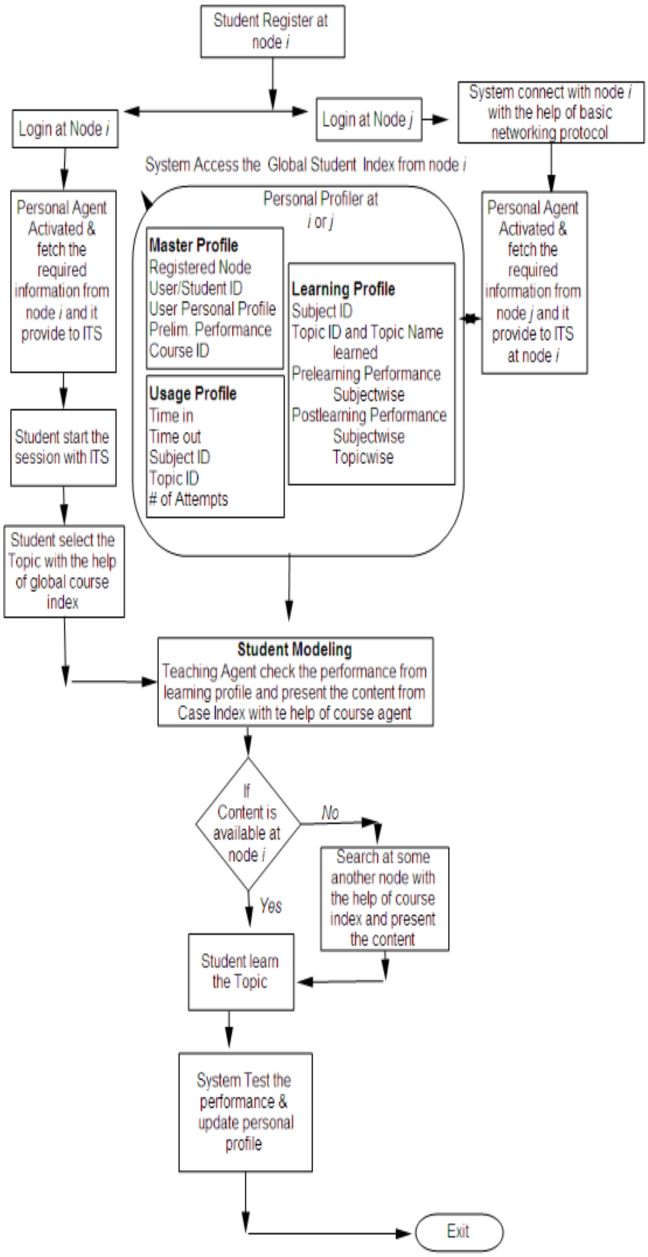
Student Register at node *i*
Login at Node *i*
Login at Node *j*
System connect with node *i* with the help of basic networking protocol
System Access the Global Student Index from node *i*
Personal Agent Activated & fetch the required information from node *i* and it provide to ITS
Personal Profiler at *i* or *j*
Master Profile
Registered Node
User/Student ID
User Personal Profile
Prelim. Performance
Course ID
Learning Profile
Subject ID
Topic ID and Topic Name learned
Prelearning Performance Subjectwise
Postlearning Performance Subjectwise Topicwise
Usage Profile
Time in
Time out
Subject ID
Topic ID
# of Attempts
Personal Agent Activated & fetch the required information from node *j* and it provide to ITS at node *i*
Student start the session with ITS
Student select the Topic with the help of global course index
Student Modeling
Teaching Agent check the performance from learning profile and present the content from Case Index with te help of course agent
If Content is available at node *i*
No
Yes
Search at some another node with the help of course index and present the content
Student learn the Topic
System Test the performance & update personal profile
Exit

Fig.3: The learning process in distributed environment

The major characteristics of the system architecture are:

1) The system is fully distributed (not bounded with any network topology) i.e. both the domain knowledge and strategic knowledge are distributed.
2) All indices are fully distributed i.e. Student Index, Case Index, and Course Index are stored on all the nodes.
3) Redundancy (duplicacy) is maintained in the storage of case base for fault tolerance. Load balancing of the cases is achieved at each node. Later one can research on optimum redundancy parameters.
4) If the node is a LAN, the domain knowledge can be localized and strategic knowledge (case base) can be global.(a school scenario, where the school is networked globally)

IV. Student Modeling in Distributed Environment

The process of student modeling [7] is shown in figure 4.

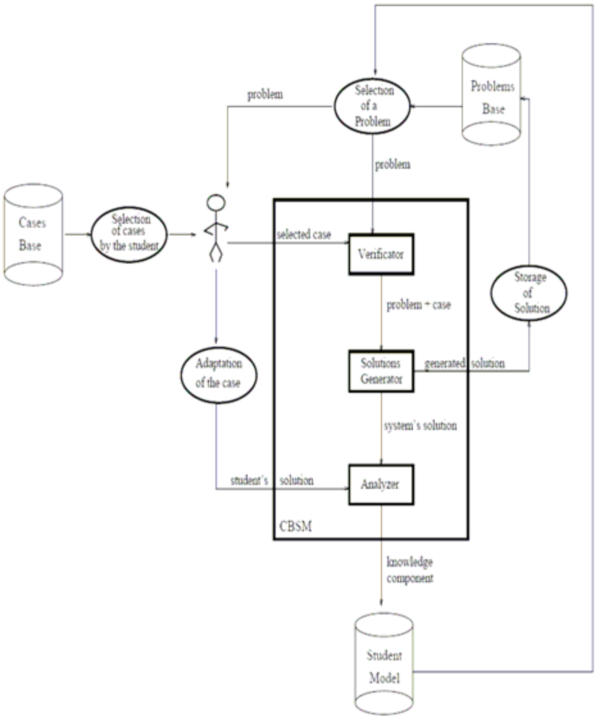

Figure: 4 The process of Student Modeling

The following activities take place in the system during the student modeling when the student interacts with the system.

1. Selection of right course material for the student and getting student's response.
2. Analyzing the student's behavior
3. Selection of appropriate case from the Case Base
4. Adaptation of the case
5. Achieving the knowledge component of the student model through case base retrieval
6. Solution generation
7. Presenting the next problem / content to the student.

In a distributed environment the complete system works as follows:

Any student can register as a user at any node over the entire network. The student can select any subject for learning. This can be done through navigation of subject provided by the ITS. The topics in a subject are required. However, before allowing for selection of a topic, the *personal agent* of the system checks the past record of the performance of the student to present the right content, thus the system is adaptable.

If the performance of the student is not satisfactory then *teaching agent* presents the appropriate remedial course content to the student with the help of *course agent*. Only after achieving the learning objectives the student is allowed to move to a new topic. If any topic (case) is not available at the local node then the system searches it from other nodes within the network with the help of course index. If the case is not found on the entire network then the *teaching agent* interacts with the human tutors through *course agent* for "plug-in topics". These new plug-in topics are stored and managed by the *course agent* on every node and at least one more node for fault tolerance. The *Course Agent* also updates the course index. This is the responsibility of the *course agent* to decide as to where should the duplication be made. The *course agent* is responsible for course management.

The design of such case based student modeling requires algorithms for the following operations:

- *Case creation in active case base.*
- *Case indexing in active case base*
- *Update case in case base*
- *Adaptation of cases in the case base*
- *Case storage Algorithm*
- *Pattern matching of case in case base or Retrieval of cases from case base*
- *Algorithm to check duplicate case from case base*
- *Case Deletion from case base and store deleted case in to archive for future reference.*
- *Student modeling*

Within the distributed student modeling, the *teaching agent* and *course agent* work for the tutor module of the ITS. The *teaching agent* does not interact directly with a student but through *personal agent* and serves as an intelligent tutor of a course. The basic components of a teaching agent are a domain expert module and pedagogy. The architecture of teaching agent is shown in figure -5.

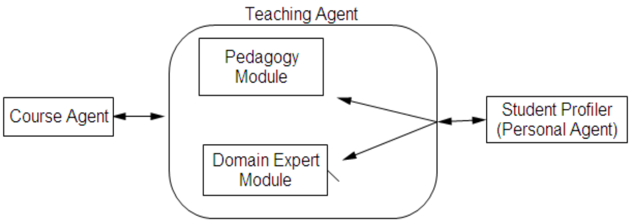

Figure: 5. Teaching Agent

The domain expert module is responsible for storing and managing the teaching strategy about learning content, exercises, assignments, and questions with the help of learning profile and *course agent.* It contains a problem generator, a problem solver, an explanation generator, and a domain knowledge base. The pedagogy module determines the timing, style, and content of the teaching agent's interventions. It is a rule-based production system that uses the student model and pedagogical knowledge to determine the appropriate actions. The student profiler provides a model of a student based on learning style, knowledge background, and interests. It may also incorporate the information about student's learning profile gathered through dialogue with the student such as the actions the student performed and the explanations which student made.

V. Conclusion

In the present work we have proposed distributed case base reasoning for intelligent tutoring system: an agent based student modeling paradigm. It uses three agents- *personal agent, course agent*, and *teaching agent*. All three agents timely interact with the system and manage the student's personal profile along with learning activities, tutoring activities, and course content management. Also it uses three indexes i.e. student index, course index and case index, which are fully distributed, to keep the record of every student, course content and stored cases by the system with in entire network.

A prototype of this distributed case based reasoning architecture for Intelligent Tutoring System is currently under development, where a 'C' Language Tutor is being designed, developed and implemented for the topic on "Operators in C". The results will be published subsequently.

Acknowledgment

The authors are grateful to the anonymous referees for their constructive and helpful comments.

References

[1] Yi Shang, Hongchi Shi, and Su-Shing Chen Deptt of computer engineering and science, University of Missouri-Columbia;An Intelligent Distributed Environment for active Learning, may 1-5,2001 hong Kong. ACM 1-5113-348-0/01/0005.

[2] Ang Yang, Ray Kemp, Kinshuk; 2001, Web based intelligent tutoring system.

[3] Sinha M. & Govil R,; 1995, PRABODH: A Distributed online Hindi Grammar Teaching Learning System. IJMS vol 11, No.3.

[4] Brusilovsky, P., Schwarz, E. Weber, G. ELM-ART: An intelligent tutoring system on World Wide Web. Intelligent Tutoring Systems Lecture Notes in Computer Science, Vol. 1086. Berlin: Springer Verlag. (1996) 261-269.

[5] Aamodt, A.; 2005, Knowledge Intesive Case-based reasoning and Intelligent tutoring system.

[6] Leen-kiat Soh; 2005, Incorporating Intelligent tutoring system in to CS1,SIGCSE06.

[7] Shiri M. E. , Aimeur A. E. and Frasson C.; 1998, Student modelling by Case Based Reasoning, Lecture Notes in Computer Science, vol 1452.

Authors Biographies

O.P.Rishi received the M.Tech. Degree in computer Science from Birla Institute of Technology, Mesra Ranchi (India) in 2000, the M.Sc. Degree in Mathematics from University of Rajasthan in 1995, He is a Ph. D. Student in Computer Science at Banasthali Vidhypith, Banasthali Rajasthan.
His current research involves Case Base Reasoning for Intelligent Tutoring Systems and Intelligent Systems.

Dr. Rekha Govil is professor, Computer Science and Dean Apaji Institute of Mathematics & Applied Computer Technology, Banasthali Vidyapith . She has more then twenty years experience of teaching and research, and has executed a number of research projects. She is into guiding Ph.D. research and till now ten students have been awarded Ph.D. Her research interest are very wide and include AI, smart instrumentation and educational computing especially in Indian languages.

Dr. Madhavi Sinha received the Ph.D. Degree in Computer Science on the topic Intelligent Tutoring System from Banasthli Vidhypith, after M.Sc. degree in Computer Science from the same university.
Her research interest is in NLP and other artificial intelligence applications.