Chapter 2: Literature Review

2.1 Introduction

In case of an institution, assessing the students’ knowledge is a very challenging task. Teachers have their own opinion, strategy and idea regarding the students’ assessment process. The students’ assessment process is very time consuming and a large no. of human resources are required. Over the last few years, the Online Assessment technologies gradually replacing the traditional paper based assessment process; thanks to the Information and Communication Technology (ICT) which widespread use has helped the Online Assessment Process.

As the proposed expert system is an agent based system and should be able to work in distributed environment, therefore I have reviewed literatures regarding Agent Technology and Agent Technology with Expert System. These literatures have given some concepts regarding the common issues associated with the design, development and the use of agent-based systems. I have discussed these, keeping the benefits and challenges of developing and using them, in mind. Here in this chapter I have reviewed some of the important theories and concepts related to conventional ES along with frameworks for developing them.
2.2 Online Assessment and Student Modelling

2.2.1 Online Assessment

In a traditional face to face learning environment, the students are assessed by conducting some kinds of unit test or term-end examinations. On the basis, how students have done in the examination, some grades are assigned to them and promotes to the next class. Over the last decade, the no. of students having aspirations for higher education is increasing year-by-year. If we consider the case with the country like India, the no. of traditional mode universities are very less and also there are private universities which squeeze more money than the government run universities/deemed universities. This leads to the evolvement of the concept of Open and Distance Learning (ODL), a huge number of students are able to opt this type of education system. This popularity lies in the fact that there is no age limit of the learners and its flexibility in learning opportunities. But there are some disadvantages in the ODL system. These are:

(a) physical communication gap among the teachers and the learners;

(b) proper assessment of students’ knowledge.

But in case of traditional classroom face to face learning or ODL system, the main problem I have observed is related with the learners’ assessment policies we adopt. The solution is to adopt online assessment techniques. The advantage of using online assessment techniques in both the face-to-face learning and in ODL systems are:
(a) Instant feedback

(b) Easy individual and group feedback

(c) Monitor group process as well as outcomes

(d) Easy distribution to markers

(e) Less time marking

(f) Increased opportunities for practice and feedback

Over the last four decades, in case of education and training, the computer has been used as an important tool. Development in the field of computer brings more and more tools for education and training. But these tools do not have that potential for competing with the real assessment sense of a teacher. This leads to the use of Expert System in educational field.

2.2.2 Student Model

While dealing with the knowledge testing of a student, there are different cognitive issues like evaluation of student's performance, identifying the misconceptions and isolating them and other various issues [1]. This leads to concept of Student Modelling. The use of the concept of student modelling has been extended to most current educational software applications that aim to be adaptive and personalised.

There are two major types of information those may be contained in a student model: model of domain-specific knowledge and model of individual, learner-related characteristics. My work is an extension of ESOA
(Expert System for Online Assessment) and it implements the Overlay Student Model (as in ESOA).

The main assumption underlying the overlay model is that a student may have incomplete but correct knowledge of the domain. The overlay model contains the student’s knowledge as a subset of the expert or domain knowledge which reflects the expert-level knowledge of the subject [2,3] (Figure 2.1). It works on the basis that students will learn the domain and gain knowledge through aspiring to become experts. The domain is decomposed into a set of elements and the overlay model is simply a set of masteries over those elements. Knowledge is represented and structured in the same way for both the domain knowledge and the student model, the difference being in terms of completeness. Knowledge representation techniques include rule-based representations and semantic networks. During student modelling, diagnosis takes place by comparing the student’s knowledge with the domain knowledge and the difference is explained as the student’s lack of skill.
This method is incomplete because only the lack of knowledge can be modelled. The main problem with the overlay model is that it assumes that a student’s knowledge can be merely a subset of that of an expert, which may not be the case. The domain model is usually represented in terms of atomic units, that is, a student either knows or does not know a certain unit. A student’s partial knowledge of a unit cannot be represented. Also, it does not represent any knowledge or beliefs, such as misconceptions, that the student might have that differ from those of the expert. That is the reason why many adaptive and/or personalized tutoring systems perform student modelling, combining overlay model with other student modelling approaches like stereotypes, perturbation and fuzzy techniques.

2.3 Software Tools for Online Assessment

A large number of software tools are available today which can be used for the assessment of student learning and these can be categorised into two classes (a) proprietary software and (b) free and open source software.

Proprietary Softwares

(a) Blackboard [4]: It is a widely used Learning Management System (LMS) all over the world. The system has a module that handles assessments, but the types of assessment offered are relatively simple ones like true/false and matching. The system also offers a proprietary programming interface which allows developers to create new types of assessments for use within Blackboard only.
(b) PLATOLearning [5]: It is a company that offers an LMS. In terms of assessment, its great strength lies in the fact that it has a library of almost 200,000 prepared problems that can be directly incorporated into classes. But the system uses some basic types of assessment techniques.

(c) SumTotal [6]: It is a well-known vendor of instructional system. Their system, TotalLMS, offers a great deal of functionality associated with an LMS including powerful assessment capabilities. But these assessment capabilities are still limited to the most basic types of problems.

(d) Wondershare Quiz Creator [7]: It is a software tool which offers flash/web based interactive interfaces for assessing student learning activities. With this assessment tool, question bank of different subjects can be designed. This software tool is very cost-effective and it is very easy to set assessment with our choice.

(e) Adobe Captivate [8]: It is one of the mostly used e-Learning authoring software. It is basically used in industries. The tool can assess learner performance by monitoring learner progress. A wide variety of interactive quizzes can be set in a very easy manner.

Free and Open Source Software

A large number of Free and Open Source Learning Management Systems are available and some of them include module or component for implementing online assessments of student learning.
Moodle is an Open Source Learning Management System (LMS), which is easily extendable and customizable [9]. It is also called as a Virtual Learning Management System (VLMS). Along with comprehensive online documentation, there is lot of support available from the active online community regarding using/customizing/extending Moodle [9]. Moodle's quiz module is one of the most complex pieces of the VLE. There are a large number of options and tools in the quiz engine, making it very flexible.

Efront is another popular open source learning management system. The system also includes an interactive module for student assessment purpose. The teacher can design individual set of self-assessment tests (multiple choice/true-false) for the learner. The student can take these tests anytime and the module provides instant feedback after each attempt of the test.

The main demerits of the assessment module of both the LMSs are:

(a) No consideration of learner's academic history,

(b) No consideration of learner's previous attempt history,

(c) In a particular self-assessment test, though each learner gets a new question randomly at each attempt, actually the questions are delivered from the set of the fixed number of questions only.

2.4 ESOA: A Review

ESOA stands for Expert System for Online Assessment. It is a rule-based expert system and it gathers two sets of factual data: (a) student academic history and his/her earlier performances, (b) a set of questions depending
on (a) [10]. The questions are categorized into different level of toughness and the academic history of students is categorized depending on some defined properties/characteristics. ESOA also contains rules for generating the question sets for students depending on their categories and for processing assessment results.

The ESOA uses JSP and Servlet technology along with JESS. The whole system is designed in MVC Paradigm. J2EE’s architecture maps onto the MVC nicely [10].

**2.4.1 Architecture of ESOA**

The ESOA architecture includes mainly five components [10]:

- Apache Web Server: listens for web page request

- Tomcat Servlet Engine: serves dynamically generated web page using JSP and Servlet technology

- MySql Database: stores the student information and the question bank of various courses.

- XML: files for external Rule Base

- JESS Engine: for online assessment
2.4.2 Working Principle of ESOA

The working principle of ESOA is as follows [10]:

- When the online assessment for a particular subject/paper of a programme is needed to start, the administrator will log-in through web-browser.

- The administrator will then request to start the pre-assessment processes with required/related inputs through web-browser.

- The servlet will handle the request and sends it to JSP which in-turn instantiates JavaBeans for retrieving question fact data from MySQL and rules from XML files and then passes them to JESS Engine.

- JESS Engine then loads the question facts, rules into its working memory, rule base and informs the administrator.
A student will have to log-in and request for the online assessment on the particular subject/paper of a programme through web-browser.

The servlet again will handle the request and sends it to JSP which in-turn instantiates JavaBeans for retrieving student fact data and passes to JESS Engine.

JESS Engine then loads the student facts into its working memory and rules into its rule base/knowledge base.

The JESS Engine then generates questions for the student based on the student academic/performance history and instantiate JavaBean to store this question set with the student identification information into MySQL database and frees memory, held by the generated question set, from the working memory.

The JESS Engine then call JSP to view the question set to the student to answer and keep track of his/her answers.

When the student completed his/her examination, JSP will instantiate JavaBean to load the question set given to the student and his/her respective answers from MySQL database to working memory of the JESS Engine and tells the JESS Engine to processes the assessment result.

The JESS Engine, after processing the assessment result, generates some feedbacks and instantiate JavaBean to store the student’s
performance/given feedbacks into the MySQL database and send the result/feedbacks to JSP for the student to view.

- The JESS Engine then frees the memory held by the student academic/performance history from its working memory.

2.4.3 Knowledge Acquisition in ESOA

Knowledge acquisition approaches generally consisting of two components: (a) interview of expert and (b) conducting literature review. [11]

In expert systems, there are two types of knowledge [11]:

(1) Factual knowledge: This kind of knowledge is the knowledge of the problem domain which is found in textbooks or journals. Factual knowledge is represented usually in most of the expert systems developed for diagnosis purposes.

(2) Heuristics knowledge: This kind of knowledge is domain specific and learned from experiences or from a mentor.

ESOA consists of three elements: (a) domain expert (b) assessment expert and (c) student model. All three elements require knowledge acquisition in some form but there is a significant difference in the timing of that acquisition. ESOA is designed with modular structure. Three modules are used in the system: (a) knowledge acquisition module (b) student/learner module (c) expert assessment module. [11]
Since my research work is related to the application of Agent Technology in Expert System, so I have reviewed various literatures related to the application of Agent Technology and Agent Technology in Expert System. In the following sections I will discuss some significant research works which gave me a way to proceed in my work in the appropriate direction.

2.5 Literatures of Agent Technology

Bellifemine and his colleagues in their paper [12] (in 1999) beautifully and structurally presented the features the Java Agent Development Framework (popularly known as JADE). JADE is a software framework to develop agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. JADE deals with all the aspects those are not related to the agent internals and are independent of the applications. These aspects include message transport, encoding and parsing, or agent life-cycle etc. This paper presents the JADE describing its intended uses. This paper also gives a walkthrough of JADE internal architecture.

Raibulet, Claudia and Demartini (in 2000), in their paper [13], discussed a case study which is related to the use of mobile agent technology for managing the distributed systems. This paper presents the implementation example of the mobile agents used to access and to manage information stored in the distributed database repository (DDR), as specified within the ESPRIT Project “Network Oriented Application Harmonization” (NOAH). Here it is mentioned that the DDR model represents a common distributed information repository for any application. According to this paper, the immediate advantage of this model is a unified perspective for several
activities that concern the distributed resources of the industrial system (e.g., installation, maintenance and technical management). In the implementation of DDR model, an object oriented database is designed and developed in the Java environment. And in the case of mobile agents’ implementation, the Voyager platform which is developed by ObjectSpace Inc. is chosen [13]. The DDR model validation has been demonstrated with an example of an actual system: a Temperature Loop in the AMPilot laboratory at ENEL in Milan [14]. In this paper it was also concluded that in the case of the management of distributed resources, the comparison between a client/server solution and a mobile agent based approach shows that mobile agent technology offers important advantages, such as flexibility and scalability of the system, load balancing, on-demand services, low traffic in the network, and many others [13].

Tripathi and his team-mates, in 2002, in their paper [15], discussed the architecture and the environment (basically the programming) of Ajanta. This is a java based system for application programming using mobile agents. In this paper, the various choices for design and solutions to problems in several areas.

It provides features for creating and dispatching agents. It also has features for controlling agents with necessary securities at remote sites and also for transferring the agents from one server from one server to another server. In Ajanta, a proxy-based access control mechanism for secure access of the intermediate servers’ resources by the visiting agents. Few of the
mechanisms in the Ajanta design are: (a) mechanisms to protect an agent’s state, and (b) prevent misuse of its credentials.

Here in this paper, the authors describe the use of migration patterns for programming an agent’s travel path. According to them a pattern encapsulates the abstract notion of agent mobility. Pattern composition allows one to build complex travel plans using some basic migration patterns [15]. They present agent-based distributed applications implemented using the Ajanta system to demonstrate Ajanta’s functional capabilities. These include a distributed calendar management system, a middleware for sharing files over the Internet, an agent-based middleware for distributed collaborations, and an agent-based network monitoring system.

The report [16], is an extended and updated version of the technical report Evaluation of Agent Platforms (Leszczyna, 2004), written in 2004, which summarised the results of our evaluation of agent platforms aiming at choosing the agent platform for the FP6/IST Personalised Information Platform for Health and Life Services project [PIPS FP6/IST eHealth Integrated Project No. 507019] and for the practical studies on security of agent systems. This report does not only contain the evaluation results but also have descriptions on mobile agent, mobile agent platform, and standard compliance. Leszczyna also presented in this report on the platforms currentness and popularity.

Lee and Hwang (in 2004), in their paper [17], discussed the possible architectural models of agent-based systems. They, in their work, evaluate
these models using metrics and then compare the architectural models based on the results on these evaluations. They had pictorially represents different organization structures (figure 2.4) which can be considered as one of the factors that affects the interaction between agents.

![Organization structures](image)

Figure 2.4: Organization structures - (a) peer-to-peer; (b) group with facilitator; (c) tree (adapted from Lee et. at., 2004)

They also discussed in their paper different coordination mechanisms [17]:

- Matchmaker: It is placed in the middle between the requester and the server (Figure 2.5). Matchmaker coordinates interaction among agents as follows: (1) server advertises its capability or role to matchmaker; (2) requester asks matchmaker to answer its request; (3) matchmaker searches the server able to answer to the request in advertised servers and replies the server's name to requester; (4) requester directly asks the server to answer to request; (5) the server directly answer to the requester's request; (6) after answering, server cancels its advertisement in matchmaker. After matching requester with server, matchmaker does not intervene between requester and server.
• Broker: Broker coordinates interaction among agents as follows (Figure 2.6): (1) server advertises its capability or role to broker; (2) requester asks broker to answer to its request; (3) broker searches the server able to answer to request and asks the server to answer to the request; (4) server replies to broker; (5) broker forwards server’s answer to the requester; (6) after processing, server cancels its advertisement. After answering, broker continues to intervene between requester and server. Here the requester can interact with server via broker.

• Contract-net: This coordination mechanism is bidding. Contract-net coordinates interaction among agents as follows (Figure 2.7): (1) manager asks bidders answer to its request; (2) each bidder decides whether it proposes to bid or not, if bidder decides to participate in
the bidding, then bidder participates in the bidding; (3) manager selects the bidder able to answer to request and notifies selection result to bidders; (4) the selected bidder answers to manager’s request.

Figure 2.7: Contract-net (adapted from Lee et. at., 2004)

In this paper [17], the performance analysis and evaluation of the different architectural models are performed in such an effective and organized manner that these can be considered in agent-based systems engineering activities.

Kim and Robertazzi in 2005, in their paper [18], discussed different fundamental models of mobile agent behaviour for mobile agent design. They also have discussed the significance of these fundamental models in terms of utility and flexibility. Mobile agents are able to move, with its state and data, around a network based on a specific plan to perform the functions like data mining, processing of network management information, delegating network control in networks etc.

According to Kim and Robertazzi, the in the modelling of mobile agent, one must consider the following novel behaviours and these are as follows:

✓ Mobile agents should be able to reside inside a host for some time,
✓ Mobile agents have a finite life-time,

✓ Mobile agents can make copies of themselves (also known as cloning),

✓ Mobile agents can be killed,

✓ Mobile agents can report the results to a central host, and

✓ Mobile agents also should be able to provide quality of service.

Apart from these above mentioned qualities, there are other qualities also those should be possessed by the mobile agents. There are two situations in which the mobile agents can be discarded and they are:

✓ Source can discard the mobile agent when it returns back to the source,

✓ And an arbitrary host (in the same network) can also discard the mobile agents.

In the paper [18], the authors have discussed features and variations on the different mobile agent models ranging from distributional assumptions to allowed mobile agent behaviour.

Fortino and his team-mates (in 2008) suggested a Java-based framework, named JIMAF (Java-based Interoperable Mobile Agent Framework) [19]. This framework relies on an event-driven, proxy-based mobile agent model with the support for MAS interoperability. This framework supports the interoperability of execution, migration and communication between
heterogeneous Java-based mobile agent systems. Here the results from the performance evaluation of MAS interoperability was carried by using JIMAF atop Aglets, Ajanta, Grasshopper, and Voyager, demonstrating that the high-level JIMAF approach offers high efficacy while maintaining overhead at acceptable levels for target computing environments. They have also compared their approach, on the basis of selected criteria, with other high-level approaches currently available. According to their experiment, overall, the JIMAF approach is more effective than the others for providing interoperability of execution, migration and communication.

Trillo et. al (2007) provides an up-to-date evaluation of existing mobile agent platforms. In this paper [20], they give the qualitative comparisons between the mobile agent platforms as well as evaluation of their performances in a variety of settings with an extensive set of experiments. The mobile agent platforms they discussed are: Aglets, Voyager, Grasshopper, Tryllian, JADE, Tracy, SPRINGS. They beautifully summarize the main features of all the mobile agent platforms. The paper gives the experimental evaluations in different ways such as Experimental Comparison with a Sample Test, Performance Comparison Executing a Parallel Algorithm [20]. The qualitative comparisons are nicely presented in the paper to provide an overview of the different alternatives for selecting a mobile agent development framework.

In the year 2009, Cucurull and others, in their article, proposed an agent migration model based on the communication standards of the IEEE-FIPA organisation. Here, the approach described encompasses the definition of
several specifications to achieve interoperability in the migration process in heterogeneous environments [21]. Their model provides a basic and extensible common migration process. This model is flexible enough to support different kinds of migration methods and future upgrades. It is completely independent of any specific middleware implementation.

The above figure 2.3, depicts the general migration process in the proposed model (mentioned above) from a high-level point of view.

![General Migration Process Diagram](adapted from Cucurull et. al., 2009)

This migration process includes the following steps [21]:

1. The agent contacts the local AMM, requesting to migrate.
2. The local AMM suspends the agent execution, and starts the ACL message exchange with the remote AMM to agree with the next sub-processes.
3. The agreed sub-processes are carried out:
   - optional agent authentication;
Chapter 2: Literature Review

- agent transfer (including agent code+data+state); and
- optional operations.

(4) The remote AMM creates and registers the new agent using the code+data+state.

(5) The local AMM shuts down the local agent and requests the remote AMM to start the agent.

(6) The remote AMM starts the new remote agent and informs about the success of the process.

In the above article, also the inter-platform mobility support of the proposed model is beautifully presented.

In the year 2013, Bahaj and Soklabi describes JADE Agent Programming Language (JAPL) that allows fast and efficient implementation of intelligent behaviours into mobile agents, based on three logical, FIPA speech acts, and a part of complex procedural script for actions in their research paper. The JAPL integrates the agent ontologies. It also defines different communication services. It also describes how to program the multiple JADE behaviours using JAPL instructions and how to compile JAPL to JAVA classes [22]. In this paper, they have given nice descriptions on the overview on JAPL, knowledge representations, agent behaviours, service concepts and its application approach. The JAPL provides constructs for describing ontologies, protocols, services and complex actions for the system of JADE mobile agents [22]. According to them, JADE agent programming language
is likely to be more fruitful than all old languages, in bridging the gap between theory and practice in the development of the JADE mobile agents.

2.6 Literatures in Agent Technology in Expert System

James G Williams and Ken Sochats (in 1996), in their paper [23], beautifully and systematically presented an overview of Expert Agents and Assistants. They also have discussed different activities related to the different areas of library and information centres. The significantly pointed out in what areas and how expert systems with the support of multi-agent system are useful in library and information centres.

According to Williams and Ken, the two main and basic features of intelligent agent software for assistance are:

- it must be able to learn, and
- it must be able to negotiate.

They basically gave a brief description on two classes of agents namely the filtering agent and the search agent. Information Filtering Agents assist in interactive browsing while Search Agents automatically search for information and these can be of enormous help with millions of WWW pages and large library collections as well as corporate databases [23]. They, by their study of OPAC uses, have found that the basic need of an expert agent or assistant is to understand the needs, intentions and goals of the users and also to develop a plan to meets these. An expert agent to aid in
searching would also need dealing with the problem of term “selection” [23].

The paper [24] presents design and development of an Agent-based Decision Support Tool (ADST) for the use in water mains rehabilitation decision making by a U.K. water company. This paper also discusses the challenges encountered in this project, and at the same time, it also describes the adoption of an emerging agent-based technology in designing the prototype and building the prototype. The challenges mostly evolves for companies which have no prior experience in applying AI-based technology to its business, or where the domain is new to the AI community. The system models, implemented as apart of the human strategy, are used in making rehabilitation decisions which embodies the expertise of the water company engineers, their consultants and their information handling models [24].

Postal and his colleagues in 2004, propose a multi-agent based expert system architecture, “MathTutor”, which integrates different formalisms to facilitate the teacher task of developing the contents of a tutorial system and at the same time to provide adaptiveness and flexibility in the presentation [25].
The system architecture of MathTutor is shown in the figure 2.8 given below. The MATHEMA [26] is a conceptual model and it is the basis of the system architecture of MathTutor. It consists of three modules and these are:

- **Society of Tutorial Agents (TAS):** It is basically a multi-agent system. In TAS, each agent has the communication and cooperation facilities. Actually, TAS can be considered as a container of a complete ITS which have the focus on a sub-domain of the target domain.

- **Apprentice Interface:** This interface provides access to the system through any Internet browser.

- **Authoring Interface:** The authoring interface module allows the definition of the course structure and contents.

In this paper, the authors have described each of the components of the architecture in a very simple, effective and technically well organized manner.
In my research work I have found many literatures related to Agent Technology. But I have found very little literatures related to Agent Technology in Expert System.

REFERENCES:


