

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## An Agent-based Architecture for Developing E-learning System

Hasan Al-Sakran

Information Technology College, Department of MIS, Yarmouk University, Irbid, Jordan

---

**Abstract:** From the perspective of e-learning view, it is important to deliver dynamic learning content and manage educational material in an efficient way. An e-learning system should present personalized optimal content to the user, anytime and anywhere and encourage collaborative learning. It should guide specific individuals to their most relevant community or communities of knowledge by exploring the materials that others in the community are viewing or reading and linking learners and with experts. The purpose of this study is to formulate a functional architecture that supports the above e-learning objectives using mobile agent technology. This architecture gives users the ability to collect, analyze, distribute and use e-learning knowledge from multiple knowledge sources. The mobile agent technology is particularly suitable for developing distributed e-learning systems because it supports intelligent and distributed storage, allows an optimal personalized e-learning environment and enhances modularity, reusability, flexibility and reliability.

**Key words:** Agent, distributed e-learning systems, teaching agent, learning agent

---

### INTRODUCTION

Most of the existing e-learning systems rely too much on the traditional learning approaches: lecture style and remote distribution and delivery of courses. These teaching approaches have major limitations, some of which are the difficulty of application of delivered knowledge; the fact that incompatible mode of delivery does not work well for all categories of students; the delivered knowledge, that is necessary to be competitive, changes very fast and may become obsolete. In recent years, numerous efforts have been made in the direction of automating the content acquisition and distribution, but personalized content delivery, access and interaction remain research challenges.

There are several different e-learning management software tools, for example, WebCT<sup>[1]</sup>. These tools offer passive and static sets of services. Instructors spend more time teaching a course on-line than the same course in the classroom, due to the time-consuming maintenance requirements of distance learning courses. Instructors are expected to check students log files, grade books, etc. They do not have the time, budget, or technical skills to build true instructional interactivity into their online learning programs. Most of these e-learning systems are designed to facilitate the work of the instructor, not to support the learning process of the students and their current needs. The diversity of the students' backgrounds and skills is ignored. The same teaching strategies are applied to students who have different profiles.

Today's e-learning systems do have special requirements. They must focus on personalization (the individual) and his or her relationships to the larger community of practitioners<sup>[2,3]</sup>. Learners must be able to interact with content in unusually complex ways (not just viewing content, text, etc). Humans or computers must control learners. That means the controller must know about the learner's capabilities, previous experiences, test results and wishes. Based on the evaluation of user's behavior, the e-learning system should make decisions of what content will be delivered next to the user. This trend is towards individual learning anywhere anytime.

Given below is the list of major requirements for an e-learning system:

- The possibility to learn anytime anywhere;
- The possibility to learn in a community (increase opportunities for interaction with other users and instructors);
- Improved the access to a vast amount of information;
- Cost effectiveness (costs can be distributed over a large number of users).
- The system must be web-based;
- It must be able to exchange and integrate information with other systems without changing these existing systems;
- It must facilitate development of new content using different technologies, such as creating a video course, publishing a research paper on the web, etc;
- It should allow reusing of the existing content if and when it is possible.

Based on the above requirements, an e-learning strategy cannot be based on the simple transmission of content.

The user of the e-learning system can be either a learner, or a supervisor. Learners do not learn on their own, but in contact with human supervisors, fellow learners, the system and the outside world. To be able to present users with the best-suited material and for administrative purposes, it is essential to track user's behavior by evaluating his behavior and career history, managing his performance and learning environment. Users will author the content as well as use it, for example make specialized material, homework assignments, classifying materials (librarians), etc.

### **MOBILE AGENT-BASED LEARNING SYSTEM**

In the client-server model communicating entities have fixed and well defined roles. A server offers a set of services and a client makes use of these services. Clients are dependent upon servers to provide the services that they require. They communicate through a message passing protocol. But message passing requires programmers to determine network addresses and synchronization points.

In the mobile agent model, when the client in the mobile agent requires access to a resource that the server provides, the client moves to the server site and makes requests to the server directly. When the transaction is complete, the mobile agent returns back to its original site with the results.

Mobile agents are intelligent computational reusable executable software on a network, which can carry out autonomous tasks. It has a degree of control over its actions and it does not always wait for commands. It adapts its behavior in response to the changing environment. An agent can migrate from server to server in heterogeneous networks. On each server, the agent interacts with stationary services and other resources to accomplish its mission. It can communicate to anticipate, adopt and plan tasks. Its behavior consists of beliefs, desires and interaction depending on the place function of an entity within an agent-based system.

A mobile agent consists of two different parts: the code itself, which composes of the instructions that define the behavior of the agent and its intelligence. The second part is the current state of execution of the agent.

To construct a mobile agent, there are at least three major requirements<sup>[4]</sup> such as:

- Common execution language across heterogeneous networks. For example Java (Due to its strengths (Portability, mobility and server security);

- Mobility of agents across networks through a communication mechanism, for example MAP, TCP/IP, HTTP, or SMTP, or other transport mechanisms;
- Protection of agents against hostile server and agent server from malicious agent. Agent may protect their data and information through encryption/decryption techniques.

Agent communication can be conducted using two methods: synchronous and asynchronous. In the first method the communicating agents must arrange for a time to communicate and must synchronize before data can be transferred. This method is used for situations where data transfer is important and needs to be confirmed. In the second method collaborating agents will communicate with each other when the receiving agent checks for received data.

The mobile agent technology seems an attractive paradigm for developing distributed e-learning systems because it supports intelligent and distributed storage allows an optimal personalized e-learning environment, helps build an intelligent decision and ensure interoperability between different systems that are to be integrated into an operational heterogeneous e-learning system. Beside that, it solves the problem of heterogeneity and low-bandwidth, reduces network traffic, process data locally instead of transmitting the data over a network. It could accelerate development using agent components and enhance modularity, reusability, flexibility and reliability.

The choice of the agent language should be compatible with the requirements needed in an educational setting. It's composed of a common protocol, a common interchange format and is adaptable to changes in the overall e-learning environment (its users, the content, educational services and hardware resources available).

### **ARCHITECTURE OF THE E-LEARNING AGENT-BASED SYSTEM**

Over the past few years, considerable research work has been conducted in using agent technology for education, for example: Pedagogical agent<sup>[5,6]</sup>, Teaching agents<sup>[7]</sup>, agents for retrieval<sup>[8]</sup>, agent infrastructure<sup>[9]</sup>. Thaiupathump *et al.*<sup>[10]</sup> shows that applying agent technology to e-learning holds promise of improving learners' satisfaction and motivation. By using such technology the learning process can be moved from human instructor to artificial agents<sup>[11]</sup>. Shi *et al.*<sup>[12]</sup> propose an intelligent solution based on a connectionist approach.

The key functions of proposed e-learning agent-based architecture are:

- Identifying optimal learning conditions through evaluating user's behavior and tracking his performance;
- Support rapid educational material development and delivery;
- Analyze personalized needs for an individual automatically and dynamically;
- Allowing agents to communicate with each other and with their environment;
- Support the interactions between sets of agents that might be geographically dispersed;
- Allow multiple agents to collaborate with each other in an asynchronous way;
- Determine what messages other agents can accept and send.

To achieve the above functional requirements, the following types of agents are needed for the proposed architecture:

- Agent Server (distributed, intelligent and multi-agent): It can resolve distributed problems through negotiation, integration and interconnection of different systems;
- Teaching agent: Assistant instructor, intelligent and self-maintaining, releases the instructor from manual monitoring of course activities. The teacher will perform the various teaching functions. Agent's owner (instructor) may configure this type of agent;
- Learning agent: Intelligent agent assisting students with specific learning needs. It acts as a smart search engine searching related resources. It also receives feedback from the student through assessment tools as well as from the student's profile;
- Mobile agent: It can migrate from one machine to another and can be executed asynchronously in an independent execution environment;
- Interface agent: The interface agents provide assistance to the user in accomplishing some simple tasks, like filtering. They learn from user's behavior, feedback and instructions and from other agents. User can use an interface agent to filter retrieved information from queries;

In order to build such a system an agent framework and a development environment are needed. An infrastructure supports the interactions between of agents that might be geographically dispersed on the Internet as shown in Fig. 1.

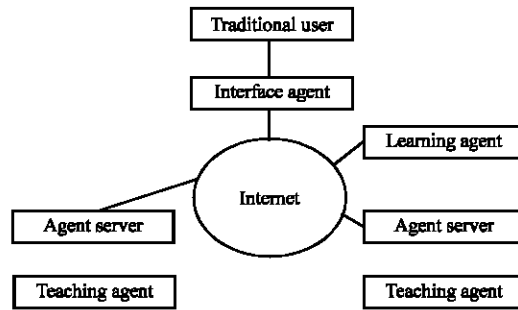


Fig. 1: E-learning architecture

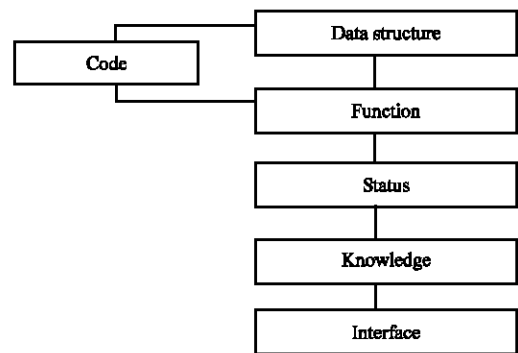


Fig. 2: Mobile agent structure

All agents in this infrastructure have common architecture regardless of their differences in functionality and intelligence. The basic functions of an agent are to:

- Communicate with other agents by exchanging messages, if necessary;
- Ability to locate and access local information services and resources;
- Migrate from server to server as needed;
- Cooperate with other agents in order to complete its job, either synchronously or asynchronously.

**Structure of a mobile agent:** The mobile agent is divided as shown in Fig. 2:

- Data structures to store the parameters of agent and retrieved information which will be sent back to user;
- Functions to satisfy users' soft and hard constraints, handle events and control agents' migration;
- Status- to store updates on agents' status and history record;
- Knowledge base to facilitate the learning process;
- Interface to facilitate communications with external environment.

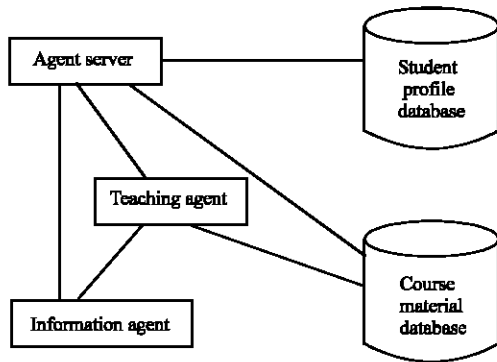


Fig. 3: An agent server

**Structure of an agent server:** The role of agent servers is to present a standard interface to other agents, managing resources to satisfy requests of the resource management agent and other agents. Figure 3 shows the structure of such an agent server.

An agent server performs the following main tasks:

- Facilitating migration of agents' code.
- Creating and maintaining of an execution environment and protection mechanisms for agents.
- Monitoring agents' actions.
- Allowing co-existence of and communications between multiple agents.
- Avoiding direct interference between agents.
- Handling agents that are coded by different programming languages by establishing corresponding language interpreters.
- Handling communications with other servers and access available services and resources through them.

**Architecture of teaching and learning intelligent agents**

**A teaching agent:** A teaching agent identifies optimal conditions in order to generate similar learning situations in the future and to avoid learning strategies that prove ineffective. It notes areas of weak achievements or knowledge, then customize the program to include material and adjust the nature and speed of the delivery of instructions. This agent fulfills several functions, such as searching and downloading related material for the student, etc. It would interact with an interface agent which in turn interacts with a course management agent. A course management agent requests learning material from the course material database and from the information agent that searches all learning resources. Figure 4 shows the structure of teaching agent.

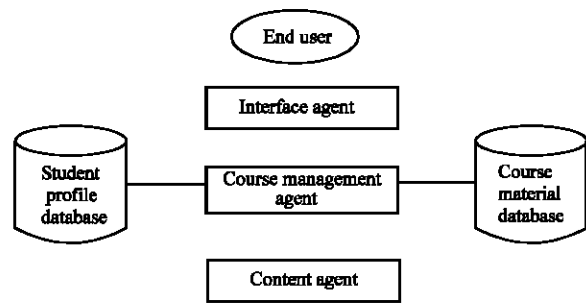


Fig. 4: Teaching agent structure

The functions of a course management agent:

- Assessment of user's profile and behavior;
- Use assessment results to associate the user with a group of people with similar preferences.
- Control the flow of content by specifying content units which are going to be presented to a user;
- Facilitate cooperation between users in order to accomplish a learner task such as distribution and submission of homework assignments.

The content agent's functions are:

- Search, retrieval and reuse of different forms of multimedia content;
- Logical content management;
- Combining logical units into a meaningful content;
- Ability to differentiate between the physical and logical organization of a collection of content units (for example: physically data can be organized as a table of relational database, logically as a scheme, so general SQL queries can access a piece of content associated with a particular logical attribute).

In order to reach an agreement between a course management agent, an information agent and an interface agent at an optimal or acceptable quality of service level, the course management agent needs to be instructed by the interface agent. An information agent will query user's records to retrieve current expertise; another agent has to query learning resources to discover the location of educational content, matching the intentions of the user.

**A learning agent:** Learning agents provide learners (students) with a personalized interface to interact with e-learning system. An end user could install one or more teaching agents for use as needed. Main tasks of an interface agent are to maintain student's profile, which contains personal data and students' suggestions and security requirements. Figure 5 shows the structure of a learning agent.

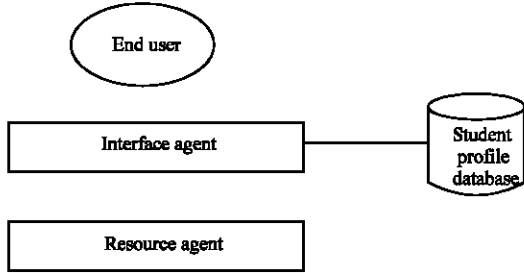


Fig. 5: Learning agent structure

Accessing student's profile to analyze students' strengths and weaknesses will improve the expertise of this agent to provide more useful resources. It will determine the gap between what learners know and what they need to know. The student profile will consist of data that are dynamically obtained from information entered by the student, his past usage of the system, or from students past assessments.

**PERFORMANCE EVALUATION**

Here, analytical comparison study have been conducted to compare the performance of mobile agent and client/server e-learning models based on equations, which will be obtained and explained later in this section. The study will explain under what conditions the mobile agent-based solution is superior to the client/server based solution for the e-learning.

In the client/server model a user can issue repeated server calls. The user's request is passed from user's station to a server where it processed. The results are then returned back to the user over the same route. If the result does not contain the desired information, then the cycle is repeated all over again with other servers. In this model the huge amounts of retrieved un-filtered data from distributed servers are processed locally to extract the required information.

In the Remote Procedure Call (RPC), the client and server open a communications channel between the client application and the server process. The RPC parameters are sent to the server process where it passed to the appropriate server procedure. The procedure processes the parameters and produces a result which is transmitted back to the client process. The RPC is synchronous; that means the client process suspends and maintaining the process state until it receive the result from the server. In the RPC-based communication, there are typically several flows between the client and server in order to perform a transaction. In a mobile agent model, on the other hand, these flows could be reduced to sending a single agent with a corresponding reduction in network traffic.

Table 1: Client-Server/Mobile Agent Model Notations

Client-Server Model Notations	
Symbols	Description
N	Number of servers
$t_{ci}$	Transfer time from client to server i
$t_{qi}$	Time taken to process query at server i
$R_{ci}$	Transmission rate of link from client to server i
$P_i$	Probability of finding relevant information at server i
$M_i$	Size of data returned to client from server i
Mobile Agent Model Notations	
$t_{ai}$	Processing time of mobile agent at server i
$t_{si}$	Time needed for agent serialization at server i
$t_{pi}$	Time to process agent at server i
$t_{di}$	Time needed for de-serialization at server i and restart
$t_{mi+i}$	Time to move mobile agent from server i to server i+1
$S_i$	Size of mobile agent after visiting server i
$P_i$	Probability of finding relevant information at server I by mobile agent
$R_{i+i}$	Transmission rate of link from server i to server i +1

The Table 1 summarizes the client/server and mobile agent models notations:

In the client/server e-learning model<sup>[13]</sup>, the client has to make a remote call to a server i to obtain the required information. Client specific requirements are then translated into many interactions between the client and servers, which affect application's performance. It may happen that a second level for each or some of the retrieved results required a remote call to N servers. Therefore the total time (TC) is:

$$TC = \sum (2t_{ci} + P_i (t_{qi} + M_i/R_{ci})) \quad (1)$$

To optimize the data transfer using a particular compression method, it is necessary to extend the server with the compression function to compress M. But the client must use the same compression function.

In the mobile agent e-learning model a user may launch an agent from his station into the network, this agent will visit different servers sequentially until it obtains the required information and then will return back to the original host. It means that the accumulated target data are moved with the mobile agent which strongly affects the size of the mobile agent. Each time a mobile agent finds a target data in a server, the size of the mobile agent is increased by the size of that data. The sequence in which servers are searched is assumed to be determined in advance, before an agent has been launched into the network. The two major parameters affecting the mobile agent performance are the mobile agent size and the time that agent requires to migrate between servers. The larger the size of a mobile agent, the more time is required to move between servers.

An agent migration between any two servers consists of the following steps:

- Agent Serialization: Construction of a message representing agent's data and code;
- Agent transfer: Sending the serialized agent (message) to the target server over TCP/IP or ATP. At the target server a thread is constructed and the message is delivered to this thread;
- Agent de-serialization: Reconstruction of the agent's data in the original form.
- Agent migration costs much more than a client/server invocation. Therefore, moving an agent to a server must save a sufficient number of remote communications between client and server in order to be beneficial.

Using mobile agent technology, the client creates an agent  $A_c$  which contains the client request to be executed. This agent moves to a specific server  $i$  where it obtains required information if available, then moves to another server where new information might be added to the data of the agent and so on until it returns to the place of its origin and delivers the final result to the client. In this scheme, the total agent time (TA) that an agent requires to migrate from the client through  $N$  servers and back to the original client is described by given below formula:

$$TA = \sum (t_{ci} + (P_i * (t_{ai} + t_{mi,i+1}))) \quad (2)$$

Where:

$t_a$  - The total processing time of mobile agent at server  $i$  :

$$t_a = t_{si} + t_{pi} + t_{di} \quad (3)$$

$S_i$  - The size of the mobile agent might increase at visited server  $i$ :

$$S_i = \sum S_j \quad (4)$$

$t_{mi,i+1}$  - Time needed to move a mobile agent of size  $S_i$  from server  $i$  to server  $i+1$  over the link between server  $i$  and server  $i+1$  with transmission rate  $R$ :

$$t_{mi,i+1} = S_i / R_{i,i+1} \quad (5)$$

To minimize the size of the data moved within the mobile agent, the agent compresses data obtained from each server and returns to the user's site with the compressed data. On the client site, the agent decompresses data and delivers it to the client. Figure 6 shows results for sufficiently large  $N$ . The results show that for a small size of data, client/server performs better than mobile agent model. The time for small data exchange dramatically reduces in the client/server model. However,

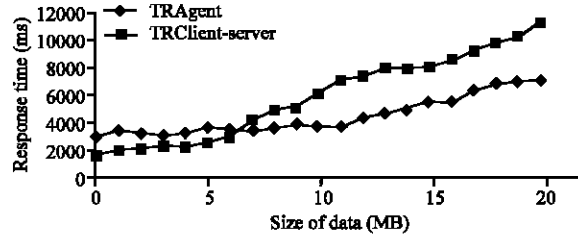


Fig. 6: Comparison between client/server and mobile agent models

for large size of data, agent performs better than client/server approach. This is due to the fact that the mobile agent-based model benefits from fast connection among servers, while client/server model needs to exchange more data frequently.

That means mobile agents based solution is much more efficient than client/server model based solution. Thus the mobile agent can utilize the bandwidth of the network much more efficiently than can be achieved by client/server model. In order to achieve better performance of mobile agent in a large distributed environment, Mawlood-Yunis *et al.*<sup>[14]</sup> propose to use multiple agents instead of single agent approach.

## CONCLUSIONS

This study proposed architecture for an e-learning system based on mobile agent technology. The mobile agent e-learning system continuously monitored the learner's actions. In the process, it performs two functions: it identifies doptimal learning conditions and noted the areas of weak knowledge.

The main advantages of the architecture are supporting the process of composing personalized content for an individual user, rapid course development and collaboration. The proposed architecture significantly increases the performance by comparison with the client/server approach, especially when the mobile agent movement allows saving communication time between the user site and the servers.

The future direction of this research will be to expand the mobile agent e-learning system to deal with wireless system.

## REFERENCES

1. WebCT Course Management Software, 2001. Web: <http://www.webct.com>
2. Hung, D., 2001. Constructivism and e-learning: balancing between the individual and social level if cognition. Educational Technol., 3: 40-44.

3. Ip, A. and S. Naidu, 2001. Experience-based pedagogical designs for e-learning. *Educational Technol.*, 9: 53-58.
4. Gray, R.S., G. Cybenko and D. Kotz, 2000. Mobile agents: Motivations and state of the art. Dartmouth College, Technical Report: TR2000-365
5. Johnson, W.I. and E. Show, 1997. Using agents to overcome deficiencies in web-based courseware. Proceedings of the Workshop Intelligent Educational Systems on the World Wide Web. 8th World Conference of the AIED Society, Kobe, Japan, pp: 18-22
6. Selker, T. and A. Coach, 1994. Teaching agent that learns. *Comm. ACM.*, 37: 92-99.
7. Marin, B. and A. Hunger *et al.*, 2004. A Generic framework for an interface tutor agent within a virtual collaborative learning environment. IEEE International Conference on Advanced Learning Technologies (ICALT'04) Joensuu, Finland, pp: 31- 35.
8. Hiltz, S.R. and B. Wellman, 1997. Asynchronous learning networks as a virtual classroom. *Comm. ACM.*, 40: 44-49.
9. Holt, P., F. Lin and G. Jelica, 2001. Infrastructure for developing agents for distance education on the internet. *J. Comp.*, pp: 77-92.
10. Traiupathump, C., J. Bourne and J.O. Campbell, 1999. Intelligent agents for online learning. *JALN.*, Vol. 3, Issue 2-November, [http://www.aln.org/alnweb/journal/Vol3\\_issue2/Choon2.htm](http://www.aln.org/alnweb/journal/Vol3_issue2/Choon2.htm)
11. Muller, J.P., M.P. Singh and A.S. Rao, 1999. Intelligent agents V: Agent Theories, Architectures and Languages. Springer-Verlag.
12. Shi, H., S. Revithis and S. Chen, 2002. An agent enabling personalized learning in e-learning environments. ACM International Conference on Autonomous Agents, Italy, pp: 847-848.
13. Arcelli, M. and A. Chinanese 1998. Client-server architectures for distributed learning environments: A proposal. 31st Annual Hawaii International Conference on System Sciences-Vol. 7
14. Mawlood-Yunis, A. and A. Nayak *et al.* 2004. Comparing performance of two agent platforms in distributed search. Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp: 425-428.