

# Agent based system for distance evaluation

Vikram Jamwal

KR School of Information Technology  
IIT Bombay, Powai  
Mumbai, 400 076, India  
vikram@it.iitb.ac.in

Sridhar Iyer

KR School of Information Technology  
IIT Bombay, Powai  
Mumbai, 400 076, India  
sri@it.iitb.ac.in

## ABSTRACT

Growth of the Internet has created new avenues for distance education. Distance evaluation is a critical component for the success of distance education initiatives. Present web based testing techniques have several limitations such as susceptibility to network delays, partitions, etc. We believe that agents can play an important role in effective structuring of such distributed applications. We have designed and implemented MADE, a mobile agent based system for distance evaluation. In this paper we discuss the architectural details of MADE.

## Keywords

Agents, Mobile Agents, Distance Education, Distance Evaluation

## 1. INTRODUCTION

The growth of distributed systems, especially the Internet, has made it possible to impart education on a large scale. A crucial factor for the success of distance education is *distance evaluation* (DE), i.e., the evaluation of students who are spread across a large distance. Some of the existing efforts towards automating the evaluation process are Computer Based Testing (CBT) and Web Based Testing (WBT).

CBT typically involves maintaining a large database of questions, which may be presented to students in an adaptive manner. Such techniques also provide for instant scoring and flexible scheduling.

WBT is an attempt to extend CBT to the Internet scale. WBT typically employs client-server architecture, eg., [1]. These evaluation systems usually use HTML-forms or java applets for student interface and either CGI-scripts or Java-servlets for back end processing. Component based approaches using Java Beans have also been explored [4].

However, the above techniques have disadvantages in the form of susceptibility to network delays and link/node failures. Also these models do not completely support features like: push-model, delivery of dynamic content, provision for evaluation of subjective questions and off-line examinations.

We believe that these features are extremely desirable for distance

evaluation and that it would be difficult to implement them using traditional client-server technologies. We propose the use of mobile agents as an alternate implementation mechanism to provide for the above features. A mobile agent (MA)[2][5] is an autonomous software entity that can migrate between various nodes of the network and perform computations at these nodes. MA carries its state information while moving from one node to another. MA has an *itinerary* (which is a list of nodes to be visited) associated with it.

We have designed and implemented MADE, a mobile agent based system for distance evaluation. We consider the full gamut of the examination process, viz. examination setting, distribution, testing, and evaluation and result compilation.

While agents have been used for a variety of purposes [6], we believe that MADE is the first attempt to use agents specifically for distance evaluation.

In section 2, we present the overview of MADE. Section 3 provides the details of design and implementation of MADE. We conclude in section 4 by presenting our observations and experiences in using mobile agents for distance evaluation.

## 2. MADE: Overview

Made encompasses the complete examination process which consists of the following three stages:

- *Examination setting:* Question paper is set by one or more examiners remotely from their individual machines.
- *Distribution and Testing:* The question paper is distributed to different examination centers and later presented to the students.
- *Evaluation and Result Compilation:* The answer papers are collected, evaluated and the results compiled.

Made was designed to support the following features (in addition to that supported by a CBT or WBT):

- **Subjective answers:** The students may be required to provide answers that are objective, written text or involving some graphical schematics. All of these cannot be automatically evaluated and would require manual help.
- **Disconnected operations:** Due to unreliable links or for security reasons, it is desirable that, after the question papers are distributed, the students work off-line for the duration of examination.
- **Push model:** A number of students may need to be evaluated simultaneously for the same set of questions. The start time for an exam may also be decided

dynamically. In such case pushing the information to the users is the best alternative.

- **Variety of delivered content:** The questions may be delivered using dynamic content in the form of audio, video-clips, or multimedia.

We have implemented MADE using the Voyager ORB[3] framework and have successfully tested it in an experimental setup on a campus network.

### 3. Design of MADE

As discussed above, the examination process is divided into following three stages:

#### 3.1 Examination Setting

In this stage (fig 2.1), the examination paper is prepared in a collaborative manner with various paper-setters setting partial question papers from their individual remote machines. The *Paper Assembler* (central coordination authority), then collects these questions and prepares a final comprehensive question paper.

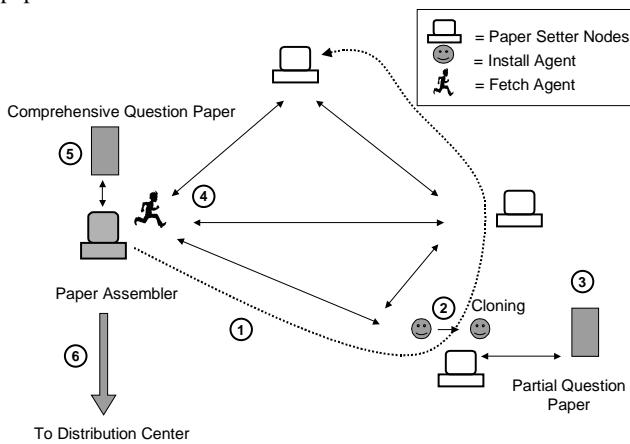


Fig 2.1 Examination Setting

This stage uses two types of mobile agents:

- **Install Agent:** This agent installs the paper-setting interface and application on the nodes belonging to the different paper-setters.
- **Fetch Agent:** This agent collects the partial papers from the paper-setters. It also enhances the paper-setter application's functionality at run-time as explained below.

The sequence of events in the examination setting is:

- i) At the paper assembler, the *Launcher* object instantiates and launches an Install Agent. This agent is supplied the itinerary which consists of the list of paper-setters that have to be visited. The Install Agent moves to the remote paper-setter and installs the remote setter's application. (Fig 2.1, step 1)

ii) The Install Agent clones itself and the clone moves to the next paper-setter. In this way the application is installed on all the machines. (Step 2)

iii) Subsequently, when it is time to collect papers, the paper assembler instantiates a Fetch Agent. (Step 4). When it arrives at a paper-setter node, the Fetch Agent creates a GUI enhancing object and attaches it to the setter's application. This allows the paper-setter to directly manipulate the Fetch Agent. This interface prompts the paper-setter to submit the questions. Depending upon the response of the paper-setter, the Fetch Agent may go into one of these states – *wait* (wait for the submission), *deferred* (move to new location and come back later), or *force-fetch* (force the paper-setter to submit the partial question paper). Once the Fetch Agent gets a partial question-paper, it moves to the paper assembler to submit it. The Fetch Agent uses a dynamic itinerary and keeps on polling the paper-setters till they have submitted their questions.

iv) The paper assembler prepares a comprehensive question paper using these partial question papers (Step 5) and then dispatches them to the *Distribution Center* (Step 6).

#### 3.2 Distribution and Testing

This stage (Fig 2.2) involves distributing the question paper to different centers, supplying the question paper to all the students, collecting the answers, and then dispatching them to the *Evaluation Center*.

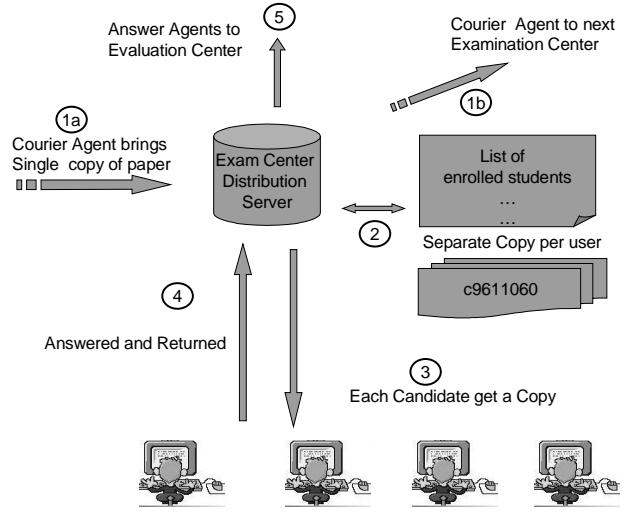


Fig 2.2 Distribution and Testing

This stage uses three types of mobile agents –

- **Courier Agent:** The main task of this agent is to deliver the question paper to all the examination centers.
- **Question Agent** The Question Agent presents the questions to the students. Once the Question Agent reaches the student machine, the student can work offline for the duration of the test. The Question Agent may time itself out after a fixed duration. When a student finishes answering the question paper or when the Question Agent times out, the Question Agent carries the answers back to the distribution server.

- **Answer Agent:** This agent represents the answer-paper of a student and moves to an *Evaluation-Center* to get its answers evaluated.

The sequence of events in the distribution and evaluation stage is –

- Upon being supplied with the itinerary and launched by the paper assembler, the Courier Agent moves to the first examination-center (Fig 2.2, step 1a). After supplying the question paper to the *Distribution Server*, it moves on to the next location (Step 1b). After finishing its task, it returns to the paper assembler.
- The distribution server distributes the question paper to all the students in the form of one Question Agent for each student.
- Once the students have answered their paper, the Question Agent returns to the distribution server with the answers. Distribution server then launches an Answer Agent, corresponding to each student, which takes the answers to evaluation center.

### 3.3 Evaluation and Result Compilation

This is the final stage in the examination process. In this stage students' answers are evaluated, and the results are compiled and published (Fig 2.3)

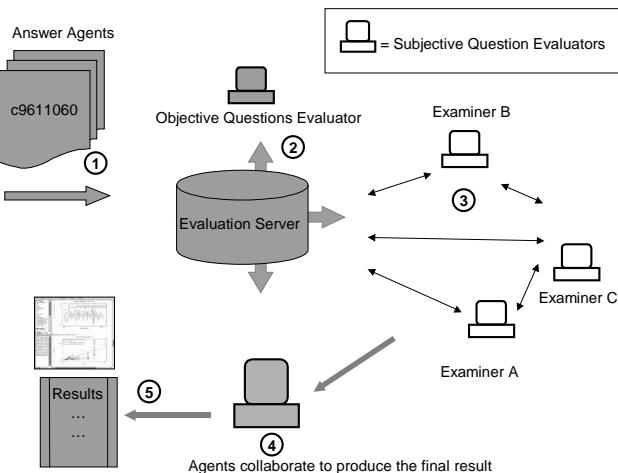


Fig 2.3 Evaluation and Result Compilation

This stage uses the Answer Agent defined in section 3.2.

The sequence of events in this stage is:

- The Answer Agent contacts (Fig 3.2, step 1) the evaluation center server to get a list of the examiners that it needs to visit. This list contains a reference to *Objective Questions Evaluator* and a list of sites of *Subjective Question Evaluators*.

- If there exist some objective answers, the Answer Agent moves to objective answer evaluator (Step 2), and gets its answers evaluated. For the rest of subjective questions it moves to the concerned examiners (Step 3).

- When all the answers have been evaluated, the Answer Agent moves to the *Publishing Center* where all Answer Agents collaborate to produce the final results (Step 4). The results are then published (Step 5).

## 4. CONCLUSIONS

We have designed and implemented MADE, a mobile agent based system for distance evaluations. We believe that such an agent based approach has several advantages such as support for subjective questions evaluation, disconnected operations, dynamic content delivery and provision for push mode of content dissemination. To the best of our knowledge, this is the first agent-based approach to distance evaluation.

We are now in the process of extending MADE to include other aspects of distance education like lecture content delivery and collaborative works.

## 5. REFERENCES

- [1] Chien Chou. Constructing a Computer-Assisted Testing and Evaluation System on the World Wide Web-The CATES Experience, in *IEEE Transactions on Education*, Vol 43, No 3, Pages 266-272, August 2000
- [2] Danny B. Lange. Mobile Objects and Mobile Agents: The Future of Distributed Computing, In *Proceedings of The European Conference on Object-Oriented Programming '98*, 1998
- [3] G. Glass, "ObjectSpace Voyager Core Package Technical Overview ", *Mobility: process, computers and agents* , Addison-Wesley, Feb. 1999
- [4] Jakob Hummes, Arnd Kohrs, and Bernard Merialdo. Questionnaires: a framework using mobile code for component-based tele-exams. In *Proceedings of IEEE 7th Intl. Workshops on Enabling Technologies: Infrastructure for Collaborating Enterprises (WET ICE)*, Stanford, CA, USA, June 1998
- [5] J. White. Mobile Agents, in *Software Agents*, J. Bradshaw (ed.), AAAI Press / The MIT Press, 1996
- [6] N. R. Jennings and M. Wooldridge. Applications of Agent Technology in *Agent Technology: Foundations, Applications, and Markets*, Springer-Verlag, March 1998