**Abstract:** The advent of networking services can support the needs for integration of management and distributed information delivery between host and destination in university environment on behalf of mankind. This paper presents mobile agent architecture for university administration system. The system performs administrative functions such as delivery orders, checking attendance and submitting reports between administrator and each department in university. The university administration system divides into two modules such as administrator module and department module in the university. The administrator module delivers orders to each department in time and then check lists of attendee from each department. The department submits reports for required information to the administrator easily and quickly using mobile agent. The proposed system is implemented with Java based mobile agent technology (Aglet) and MySQL database. The benefits of implementing the proposed system are to reduce the human errors when using the manual administration system, to be beneficial to information collection and monitoring and to give user satisfaction when utilizing the proposed system. The system aims to promote the existing traditional administration system to the computerized system.

**Keywords:** Administration System, Aglet, Management, Mobile Agent.

**I. INTRODUCTION**

Mobile-agent based system will provide effective and flexible mechanisms for structuring distributed systems. In this paper, the university administration system is implemented with mobile agent assists the user without bothering him, and to learn effectively. Mobile agents can move from one system to another within the network and do not bound to the system where it begins execution. Mobile agent contains three components. These are code that is the program (in a suitable language) that defines the agent behavior, state that is the agent’s internal variables etc., and attributes that is information describing the agent, its origin and owner, its movement history, resource requirements and authentication keys. The main purposes of the administration system are integrating administrative office and departments, and management of the functions among administrator and staffs. In the past, the traditional administration systems performed and solved the problems for managements of the functions with the cooperation of humans.

Today, most of the administration systems are developed using mobile agent technology, instead of human beings. The computerized administration system can provide an effective way to fulfill the needs of administrative processes than contact negotiation. The common university administrative approach consists of the staff handling and transmitting to each department to get required data and information about the functions and staffs’ activities. The proposed system aims to map the real world administration system and focuses on the full steps of three administrative tasks such as checking attendance, delivery orders and submitting reports. The system provides to save time and reduce human working energy. This paper is organized as follows. Section II presents the related works, Section III introduces the background theory. Section IV describes the architecture of the proposed system. And then, session V presents the implementation of the proposed system whereas section VI draws conclusions.

**II. RELATED WORKS**

Zoran Budimac [1] presented lessons learned from the implementation of a workflow management system using mobile agents. Their solution was highly distributed and almost without centralized control. Therefore, it had better characteristics than other corresponding designs, including ones that also use mobile agents as the infrastructure. They concentrated on the experience that they gained during the implementation of the system. The highlights of their system are: the changes in the original design that are enforced by the choice of underlying mobile-agent system, distribution of responsibilities among entities of the system, and possibility to use stationary service agents. The experience of using their system is positive. As planned, it is robust, highly distributed, and useful. The work on their system continues with implementation of more specialized agents and utilities for user-friendly administration of agents. Santosh Ramakrishna [2] presented an agent-based tool for system administration (ABSA). ABSA was introduced to minimize the cost of
administering computers in multi platform networks and to provide a simple, consistent, expandable, and integrated system administration tool. The agent technology in their architecture, was augmented by expert system capabilities, demonstrated a remarkable capacity for managing complexities and producing satisfactory results. By using agent technology, their system was capable of distributing the administrative tasks among the computer in the network and prevented the concentration of the tasks on a central computer. This gave the system scalability and more reliability.

Jonathan Dale [3] developed mobile agent architecture for distributed information management (DIM). They advocated the integration of two technologies; distributed information management principals to allow users to create, disseminate, discover and manage their information, and mobile agent technology to provide the flexibility, scalability and dynamism necessary to develop such distributed information management applications. They presented four principles that they has identified as being key to achieving distributed information management and an architecture where mobile agents can move across distributed environments, integrate with local resources and other mobile agents, and communicate their results back to the user. They also described a number of prototype distributed information management agents that they had developed. Future directions of their research will lie in developing a high-level scripting language in which users can develop and combine DIM agents from hierarchies of primitive sets. They were also investigating the use of Knowledge Query and Manipulation Language (KQML) to represent the specification of the communication protocol between agents and Knowledge Interchange Format (KIF) to ease information conversion between different agent systems.

III. BACKGROUND THEORY

This section briefly describes about the mobile agent, and presents the life cycle model of a mobile agent.

A. Mobile Agent

An agent is a program that assists people and acts on their behalf. Agents function by allowing people to delegate work to them. An agent possesses the following mandatory properties:

- **Reactive**: senses changes in the environment and acts according to those changes.
- **Autonomous**: has control over its own actions.
- **Goal-driven**: is proactive.
- **Temporally continuous**: is continuously executing.

And also, mobile agent may possess any of the following orthogonal properties:

- **Communicative**: able to communicate with other agents.
- **Mobile**: can travel from one host to another.
- **Learning**: adapts in accordance with previous experience.
- **Believable**: appears believable to the end user[4].

There are two types of agents, namely stationary agent and mobile agent. A stationary agent executes only on the system where it begins execution. A mobile agent can move from one system to another within the network. It is not bound to the system where it begins execution. Mobile agent has the following properties:

- The task can be automated, and delegated to a software system.
- The task is very large, and modularization is possible.
- The information needed is vast, and/or widely distributed, as with the Internet.
- The application or service needs to learn and improve with time, or be customized for each user [5].

Mobiles agents are programs that execute in contexts called places and can autonomously travel from place to place resuming their execution there. They can move freely across places in different computers and devices. The only requirement is to have a certain mobile agent platform that allows their execution in those computers. Thanks to their mobility, mobile agents offer many interesting benefits, such as encapsulating tasks, reducing the network load, overcoming network latency, executing asynchronously and autonomously, and adapting dynamically in their environment [6].

B. Life Cycle of a Mobile Agent

Figure1. The lifecycle model of a mobile agent.

The life of a mobile agent is modeled with the stages it goes through as shown in figure 1. The stages of the model are:

- **Creation** of the agent is done only once new agent is created. Every agent gets its unique id, initial state and then it is prepared for further instructions.
- **Starting** is done each time when the agent arrives to a new host. The agent has its own thread of execution and can execute asynchronously.
- **After deactivation**, the agent stops all its calculations and stores its state and intermediate results to a disk.
- **Disposal** means that the agent terminates all its activity and frees all resources it is using. After that, its state is lost forever.
- **Cloning** is a process of making multiple clones of an agent using object serialization. New cloned agent is identical same with the original except the id
Mobile Agent Architecture for University Administration System

IV. ARCHITECTURE OF THE PROPOSED SYSTEM

This session presents the mobile agent architecture for university administration system by using the advantages of mobile agent technology. The overview of the proposed system is illustrated in figure 2. The system can perform three administrative functions: checking attendance, delivery orders and submitting reports. The proposed system aims to integrate administrative office and each department in university.

![Overview of the Proposed System](image)

Figure 2. Overview of the Proposed System.

In the proposed system, there are two main sites: administrator site and department sites. The proposed system uses two stationary agents (Admin Agent and Department Agent) and three mobile agents (Attendance Agent, Order Agent and Report Agent). Firstly, at the administrator site, the Admin Agent is a stationary agent and can perform two functions such as checking attendance list of each department and delivery orders to each department in university. It creates Attendance Agent to get list of attendee from each department and dispatches it after getting the itinerary of each department. And then, it creates Order Agent to deliver orders about the activities of university to each department and dispatches it after getting the itinerary of each department. And also, it receives and records attendance data carried by Attendance Agent and reports carried by Report Agent.

At the department site, Department Agent is a stationary agent and can perform submitting reports to the administrator. It creates Report Agent to send reports about required information for the department and dispatches it to the administrator after getting the address of the administrator. And then, it accepts and records orders carried by Order Agent. And also, it gives attendance list of its department to Attendance Agent. The roles of agents involved in the proposed architecture are described in Table 1.

<table>
<thead>
<tr>
<th>Agent</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin Agent</td>
<td>- is a stationary agent at the administrator site</td>
</tr>
<tr>
<td>Attendance Agent</td>
<td>- Create Attendance Agent and Order Agent, and dispatch them to the department</td>
</tr>
<tr>
<td>Report Agent</td>
<td>- Accept and record orders carried by Report Agent</td>
</tr>
<tr>
<td>Department Agent</td>
<td>- is a stationary agent at the department site</td>
</tr>
<tr>
<td>Attendance Agent</td>
<td>- Migrate between administrator and department sites</td>
</tr>
<tr>
<td>Order Agent</td>
<td>- Request to give attendance list to Department Agent</td>
</tr>
<tr>
<td>Report Agent</td>
<td>- Accept the reports from Department Agent and move to the administrator</td>
</tr>
<tr>
<td>Report Agent</td>
<td>- Submit reports the Admin Agent</td>
</tr>
</tbody>
</table>

TABLE I: ROLES OF AGENTS FOR THE UNIVERSITY ADMINISTRATION SYSTEM

V. IMPLEMENTATION OF THE PROPOSED SYSTEM

The university administration system is implemented with Java based mobile agent technology (Aglet) and MySQL database. This session describes some of the interfaces of the proposed system. As shown in figure 3, the main window of administrator is displayed by Admin Agent at the administrator site. The administrator must click the corresponding buttons, for checking attendance, for delivery orders, for viewing attendance records, and for viewing reports. The administrator must click “Check Attendance” button if he wants to check the attendance of each department. The administrator creates and dispatches Attendance Agent to the department at that time. The administrator must click “Deliver Order” button, if he wants to deliver orders to each department. The administrator creates and dispatches Order Agent to the department at that time. The administrator must click “View Attendance” button.
if he wants to view attendance records of each department. And also, the administrator must click “View Report” button if he wants to view reports submitting from each department.

Figure 3. Main window of administrator.

Attendance Agent is created and dispatched to the department as shown in figure 4. After the Attendance Agent had been created, the itinerary of the department is selected and added. And then, by clicking “Move” button, the administrator dispatches Attendance Agent to the department for getting lists of attendance. As shown in figure 5, the main window of civil department is displayed by Department Agent at each department. The authorized person from the department must click the corresponding buttons for submitting reports, and for viewing orders. The authorized person must click “Submit Report” button if he wants to submit reports to the administrator. Report Agent is created and dispatched to the administrator at that time. And also, the authorized person must click “View Order” button if he wants to view orders delivered from the administrator. In figure 6, Attendance Agent requests to give the list of attendance to Department Agent when it reaches at the department site.

Figure 4. Creating and dispatching Attendance Agent to the department.

Figure 5. Main window of civil department.

Figure 6. Arriving Attendance Agent to civil department.

The user can add his attendance to list by clicking “ADD LIST!” button as shown in figure 7. Then, Attendance Agent is given the attendance data by clicking “SEND” button. After getting Attendance records, Attendance Agent returns to the administrator and gives data to Admin Agent. And then, Admin Agent accepts and saves data in database.

Figure 7. Giving the attendance list to Attendance Agent.
Figure 8 shows the attendance record of civil department carried by Attendance Agent. This result can be seen by clicking “View Attendance” button in main window of administrator (figure 3).

![Attendance Records](image)

Figure 8. Attendance record of civil department.

To perform delivery order function, the administrator chooses the order file to deliver to the department. After choosing the order file, the administrator creates the Order Agent and selects the itinerary of the department. Then, Order Agent is dispatched to the department. When the Order Agent arrives at the department, it gives the order to Department Agent. Department Agent accepts and saves the order. To perform submitting report function, the user from the department chooses the report file to submit to the administrator and creates the Report Agent. After creating the Report Agent, the address of the administrator is selected. Then, Report Agent is dispatched to the administrator. When the Report Agent reaches at the administrator, it gives the report to Admin Agent. Admin Agent receives and records the report. The associated tasks of Admin Agent can be seen in the status results as shown in figure 9.

![Status Results](image)

Figure 9. Associated tasks of Admin Agent.

The associated tasks of Department Agent can be seen in the status results as shown in figure 10.

![Status Results](image)

Figure 10. Associated tasks of Department Agent.

VI. CONCLUSIONS

In this paper, mobile agent architecture for university administration system is presented. The proposed system uses the mobile agent technology to present good efficiency. The system aims to enhance traditional administration system to the computerized administration system. By developing the proposed system, it gives advantages such as, implementing a convenient and efficient university administration system, achieving the fast and effective ways to perform administrative tasks (delivery orders, checking attendance, and submitting reports) between administrator and each department on university level, and saving time and reducing human working energy. By using mobile agent for implementing the system, we can get several benefits such as dynamic information delivery, reducing the network latency, and saving the network bandwidth.

VII. ACKNOWLEDGMENT

First of all, the author is highly grateful to Dr. Myint Thein, the Pro-Rector of the Mandalay Technological University for his permission for completion of this paper. The author wants to express her gratitude to Dr. Moe Moe Aye for her help and advice regarding of this topic and excellent guidance, valuable suggestions and advices. The author is deeply thankful to Dr. Aung Myint Aye, Head of Department and all teachers from the Department of Information Technology at Mandalay Technological University for their overall supporting during the writing of this paper.

VIII. REFERENCES


International Journal of Scientific Engineering and Technology Research
Volume.03, IssueNo.13, June-2014, Pages: 2807-2812