Architecture and Collaborations among Agents in Mobile Educational Game

Chris Lu, Maiga Chang, Kinshuk
School of Computing and Information Systems
Athabasca University
Athabasca, Canada
chrischien630@gmail.com, maigac@athabascau.ca, kinshuk@athabascau.ca

Echo Huang, Ching-Wen Chen
Dept. of Information Management
National Kaohsiung First University of Sci. and Tech.
Kaohsiung, Taiwan
echoh@ccms.nkfust.edu.tw, chingwen@ccms.nkfust.edu.tw

Abstract - We describe a multi-agent based context-aware mobile educational game that can generate a series of learning activities for users doing on-the-job training in their working environment. In this paper, we reveal the multi-agent architecture (MAA) that is designed in the mobile educational game. The multi-agent architecture design enables development of a lightweight, flexible, and scalable game on the platform with limited resources such as mobile phones. Six agents have been implemented that worked together in this game. Each agent has its own tasks and sometimes needs help from other agents. Using agents in developing system can not only solve the limited resource issue, but also provide us a way to add further functions and to replace agents into the game easily to improve the game anytime. This paper focuses on designing the tasks that each agent needs to do and the collaborations that may happen among agents.

Keywords - context-aware; knowledge structure; multi-agent system; mobile phone; educational game

I. MOTIVATION AND OBJECTIVES

This research is about developing a mobile educational game for employees doing on-the-job training in their workplaces. Mobile phones have limited computing power and resources compared to desktop and laptop computers, the mobile applications hence are usually small and simplified. [5] propose five design principles for developing applications on mobile devices: multiplatform adaptation, little resource usage, little human/device interaction, small data communication bandwidth use, and no additional hardware. The multi-agent based system design approach is geared towards helping in complying with the abovementioned five design principles.

Agent is an independent computer program, capable of acting autonomously and learning continuously to meet its design objectives [2]. The agents in a multi-agent system are responsible for their own tasks and collaborate with other agents whose responsibilities are helpful to them. Researchers have applied multi-agent concept in learning management system and mobile educational system design, and have reported good results in terms of system scalability [3][7].

[1] summarize the benefits of multi-agent system as: increased speed and efficiency due to parallel computation and asynchronous operation; increased reliability and robustness since it is unlikely that all agents will fail at the same time; increased scalability and flexibility since agents can be added anytime as necessary; reduced computational and communication costs due to the non-centralized architecture; and, reusability since agents can be easily replaced or upgraded.

Since not all agents are needed to be started at the very beginning, the multi-agent based educational game on mobile phones, developed in this research, consumes relatively fewer resources than on personal computers. Beside the "resource usage" design principle, DBAccess and MapHolder agents are designed in the way that the mobile educational game does not require network bandwidth all the time. With these two agents, the game complies with the design principle - "small data communication bandwidth use."

The rest of this paper is organized as follows. The system model and the multi-agent architecture of the mobile educational game are described in Section 2. Section 3 explains how agents collaborate with each others in the game-play. The game and the game-play are introduced in Section 4. Finally, the paper is concluded with a summary and the discussions of possible future research.

II. MULTI-AGENT ARCHITECTURE

The multi-agent architecture was designed while implementing the mobile educational game in order to comply with the five design principles proposed by [5] as well as to develop a lightweight, flexible, and scalable mobile education game. This architecture not only enables different agents to work on different tasks, but also provides a way for easy improvements and maintenance of the game. For instance, we can put new agents into the game for special purpose and replace old agents with new and more functional ones. Figure 1 shows the multi-agent architecture of the proposed mobile educational game.

Two groups of agents reside on user's mobile phone; three agents form a group to serve and to interact with the user, namely Player Agent, Translator, and Learning Activity Item Collector. Another five agents form a group to work out context-awareness and location-based learning activity chain,
namely Learning Activity Generator, Calculator, Map Holder, DB Access Agent, and Position Locator.

![Multi-Agent Architecture of the proposed mobile educational game](image)

Figure 1. Multi-Agent Architecture of the proposed mobile educational game

The agents have different responsibilities and tasks:

- **Player Agent** - Player agent is a bridge between the user and other agents. It gets the decisions the user made and acquires data from other agents such as Translator and Learning Activity Generator.

- **Learning Activity Item Collector** - Learning Activity Item Collector helps the user scan the QR code with the built-in camera, and decrypts and interprets the QR code. The QR codes store not both positioning data and the knowledge and instructions that the corresponding objects may have.

- **Translator** - Translator can identify different language inputs and encode/decode the inputs with appropriate character set. Translator is very useful in non-English speaking country (e.g. China, India, and Japan) as well as bilingual environments (e.g. English-French and Dutch-English).

- **Calculator and Learning Activity Generator** - The two agents accomplish the tasks of context-awareness learning activity generation. Calculator is responsible for measuring learning objects' information values according to the chosen role, theme, and the context surrounding the user. Learning Activity Generator, on the other hand, is responsible for generating activities based on the characteristics and corresponding learning objects filtered by Calculator and resorting those activities into a chain.

- **Position Locator** - Position Locator is responsible for detecting where the user is. The GPS-enabled Position Locator gathers the GPS data packets from the GPS receiver and extracts the longitudes and latitudes from the packets. On the other hand, the Camera-enabled collector gets encoded data by scanning the 2-dimension barcode and decodes the data stored by the 2-dimension barcode.

- **Map Holder** - Map Holder always keeps a copy of the map where the user is for serving other agents when the network connection is no longer available and DB Access Agent is not able to connect to the database. The map in the proposed game is a context-awareness knowledge structure, namely Ubiquitous Knowledge Structure, proposed by Wu et al. [6].

- **DB Access Agent** - DB Access Agent uses appropriate data manipulation language (i.e., SQL commands) to access data from the database for other agents. If the network connection is not available, the agent will tell other agents to look for Map Holder's help but keep those jobs requiring database update. DB Access Agent will do batch update when the network connection is recovered.

### III. Collaborations among Agents

In this section, the details of the collaboration among agents are discussed. Figure 2 illustrates the relations among the agents and database. Agents communicate with each other by sending requests and receiving responses. An agent is initiated only when other agents need certain agent’s help. With such flexible collaboration mechanism, it is possible to extend and enhance the game anytime very easily by adding new agents or replacing old ones without changing the main program.

Following paragraphs describe the stage by stage details of the collaborations among agents.

#### A. Sign in/Register

At this stage (Stage 1), four agents are involved: Player Agent, Translator, Database Access Agent and Map Holder. Player Agent first gets username and password from the user and then sends these data to Translator for checking the language used by these data. The Player Agent then sends the username and password to DB Access Agent for verification. DB Access Agent checks if the username already exists in either the game database or other system databases (e.g. Moodle database). If the username does not exist, DB Access Agent informs the Player Agent. The Player Agent then asks the user to create an account for playing the game. Map Holder is a backup of DB Access Agent.
Agent; it helps other agents to retrieve required data under offline mode (i.e., when no wireless connection is available). Figure 3 shows the whole process and communication among agents at this stage.

**B. Role and Theme chosen**

At this stage (Stage 2), two agents are involved: Player Agent and Learning Activity Generator. After the user has signed in or registered a new account successfully, the Player Agent offers the user two roles for playing, namely new employee and visiting scholar. Each role has some pre-defined themes for the user to pick-up. The chosen role and theme are then sent to Learning Activity Generator. Figure 4 shows the whole process and communication among agents at this stage.

**C. Positioning**

At this stage (Stage 3), three agents are involved: Learning Activity Generator, Calculator, and Position Locator. Before Calculator can start to filter learning objects and corresponding characteristics for Learning Activity Generator to generate learning activity chain, Calculator needs to know where the user is right now. Calculator asks Position Locator's help in getting longitudes and latitudes of the user's position from GPS data packet. Some users' mobile phones may not have built-in GPS receiver. In such cases, Position Locator can still get location data by working with Learning Activity Item Collector. Learning Activity Item Collector can scan QR Code and extract the location data stored in the QR Code. Once the Learning Activity Item Collector got location data, it passes the data back to the Position Locator and the Position Locator sends the data back to Calculator as shown in Figure 5.

**D. Learning Activity Generation**

At this stage (Stage 4), four agents are involved: Calculator, Learning Activity Generator, DB Access Agent and Map Holder. Once the user chose his/her preferred role and theme, Learning Activity Generator sends the chosen role and theme to Calculator and asks Calculator to return those learning objects and corresponding characteristics back.

Calculator first asks DB Access Agent to provide qualified learning activity templates, learning objects, and corresponding characteristics for the chosen role and theme. Next, Calculator eliminates those irrelevant learning objects and characteristics based on the location data it got back from the Position Locator at Stage 3, for instance, those learning objects outside the city where the user currently is. Finally, Calculator measures the information values of the remaining learning objects and corresponding characteristics according to the algorithm described in [4] and sends the identified learning activity templates, learning objects and corresponding characteristics with information values back to Learning Activity Generator.

After Learning Activity Generator gets data back from Calculator, the Learning Activity Generator tries every possible learning activity instances by putting learning objects and characteristics into the learning activity templates. Eventually, the selected activities are sorted based on the method described in [4]. Again, Map Holder here acts as an offline version of DB Access Agent and serves other agents such as DB Access Agent. Figure 6 shows the whole process and communication among agents at this stage.
E. Users doing learning activity

At this stage (Stage 5), five agents are involved: Learning Activity Generator, Player Agent, Learning Activity Item Collector, DB Access Agent and Map Holder. In the beginning, Learning Activity Generator gives Player Agent the learning activity chain. The user then receives the learning activities one by one offered by Player Agent. Each learning activity asks the user to find specific learning object(s) and to collect the learning objects by scanning its QR Codes. Therefore, the Player Agent helps the user to wake Learning Activity Item Collector up when the user wants to collect the learning activity item. The Learning Activity Item Collector can start the built-in camera in order for the user to take photo of QR Code, and decode the QR Code for the user. The Player Agent then checks whether the learning object collected by the user is what the learning activity asks for or not. Finally, the user's experiences are saved back to the database as shown in Figure 7.

Figure 7. The agents collaborations at Stage 5

IV. THE GAME

In this section, we demonstrate the game-play with screenshots. During the game-play, the Player Agent is the only agent who interacts with the user and helps data exchanges between the user and other agents. Correspondingly, DB Access Agent is the only agent designed to access the database to reduce the complexity of system development.

Figure 8. Screenshots of the game-play

In this game, two roles and corresponding pre-defined themes are designed that the user may choose to play. For instance, a visiting scholar may have interest in the lifestyle of the new environment s/he has just arrived in and a new employee may want to know people's offices and specific working procedures (as shows in Step 2 in Figure 8). Once the Learning Activity Generator sends the learning activity chain to Player Agent, the Player Agent provides the activities to the user one by one as shown in Step 3 in Figure 8. The learning activity may require the user to collect more than one learning activity items via the built-in camera (as shown in Step 4 and Step 5 in Figure 8). The user can review the learning content contained in the QR Code while collecting the required learning activity item as shown in Step 6 in Figure 8. Beside the text-based learning contents, the learning contents can also be HTML-based, binary-based image data, URL of webpage, media stream and Flash animation for delivering different types of information/knowledge to the user. Finally, as shown in Step 7 in Figure 8, the Player Agent checks whether all required learning activity items have been collected as well as whether all learning activities have been completed.

V. CONCLUSION

In this paper, we present a multi-agent-based mobile educational game. By using multi-agent concept in developing the mobile education game, we successfully comply with the five design principles of mobile application development proposed by [5].

Regarding "multiplatform adaption" design principle, we use Java as the programming language to implement most of the agents and use native programming languages to implement those agents which access low-level features and functions that mobile phones provide, e.g., built-in camera. Beside the programming language, we use QR Codes to store positioning data in order to make sure that users can play the game with the mobile phones without built-in GPS receiver.

Regarding "little resource usage" design principle, the multi-agent based educational game we developed on mobile phones consumes relatively less resource since not all agents are needed to be started at the very beginning.

Regarding "little human/device interaction" design principle, in the proposed game, the Player Agent only talks to users for asking them to pick role and theme at the beginning. After that, the Player Agent gives the users a learning activity to solve and only interacts with the users again after they have asked Learning Activity Item Collector to scan the QR Codes for them.

Regarding "small data communication bandwidth use" design principle, DB Access Agent and Map Holder are designed in such a way that the mobile educational game we developed does not require network bandwidth all the time. Moreover, we use QR Codes to store the knowledge and
instructions that the corresponding objects may have and to reduce the system accessing the learning contents on the Internet all the time.

Regarding "no additional hardware" design principle, similar to what is done for the "multiplatform adaption", the users are neither required to have the mobile phones with built-in GPS receiver nor RFID reader for getting longitudes and latitudes where the users as well as the learning contents are associated with specific learning objects.

Current implementation of this game can help users in on-the-job training in order to get familiar with the new environment, to adapt new working procedures and policies, and to learn physical functions related to their jobs. The multi-agent architecture makes it easy to maintain and enhance the game at anytime in the future.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support of NSERC, iCORE, Xerox and the research related gift funding provided to the Learning Communities Project by Mr. Allan Markin.

REFERENCES