AGENT-ORIENTED MIDDLEWARE SUPPORTING DELIVERY OF MOBILE ELEARNING SERVICES

Ivan Minov, Damyan Mitev

Abstract: *Within project Distributed eLearning Center (DeLC) we are developing a system for distance and eLearning, which offers fixed and mobile access to electronic content and services. Mobile access is based on InfoStation architecture, which provides Bluetooth and WiFi connectivity. On InfoStation network we are developing multi-agent middleware that provides context-aware, adaptive and personalized access to the mobile services to the users. For more convenient testing and optimization of the middleware a simulation environment, called CA^3 SiEnv, is being created.*

Keywords: Context-Aware and Adaptable Architectures, Agent-Oriented Middleware, Simulation Environments, Mobile Services, eLearning, InfoStation-Based Networks.

2010 Mathematics Subject Classification: 97R40

1. Introduction

This paper presents a context-aware and adaptable middleware supporting delivering mobile eLearning services over an InfoStation-based communication environment. The InfoStation network, which operates across a University Campus and provides access to a variety of mobile eLearning (mLearning) services and supplementary communications services [1-3]. It facilitates efficient mobile access to services and resources via intelligent wireless access points (called InfoStations) deployed at key locations around a University Campus. The system architecture consists of three tiers: *1st tier* - the user mobile devices (cell phones, laptops, PDAs), equipped with intelligent agents acting as Personal Assistants to users; *2nd tier* - the InfoStations, which facilitate the users' mobile access to services through Bluetooth and/or WiFi wireless connections, create and manage user sessions, provide interface to *global* services offered by an InfoStation Center, and host *local* services; *3rd tier* – the InfoStation Center, the core of the overall architecture, which is concerned with controlling the InfoStations, and with the overall updating and synchronization of information across the system. A set of mLearning service prototypes have been developed that use the resources distributed across this system architecture in an intelligent and efficient manner. For instance, the user requests for most of services are satisfied directly by the InfoStations without a need to access the InfoStation Center's resources (which allows reducing the workload on the InfoStation Center). Only a small portion of user requests for services would need to be redirected to the InfoStation Center in order to be fully satisfied.

In order to support the delivery of eLearning service via the above described communication network a multi-layered software architecture has been developed, called Context-Aware and Adaptable Architecture (CA^3), consisting of three level: Middleware Activation Level, Middleware and Service Interface Level [4]. The rest of the paper is organized as follows. Section 2 describes the middleware in some detail. Section 3 presents briefly the simulation environment under development for testing the middleware. Finally section 5 concludes the paper.

2. Agent-oriented middleware

Middleware software with specific functions has to be developed and used in this architecture to ensure sufficient flexibility, adaptability, intelligence and autonomy of the system architecture's nodes. In addition, the autonomic software components must be able to communicate at a higher semantic level with regard to the context, business-logic of the provided service, and individual characteristics of users. For this reason, an agent-oriented approach has been chosen for the development of the required middleware supporting a context-aware provision of mLearning services [5]. The main implementation challenges for the middleware are related to the support of distributed control, which to be able to detect all relevant changes in the environment (context-awareness) and according to these changes to facilitate the service offerings in more flexible, efficient and intelligent way (adaptability). The middleware is implemented as a set of cooperating intelligent agents (multi-agent system) in order to:

- Model adequately the real distributed communication infrastructure;
- Allow for realization of distributed control models;
- Ensure pro-active middleware behavior;
- Use more efficiently and intelligently the information resources spread over different InfoStations.

Moreover, this agent-oriented architecture could be easily extended with new agents (if necessary), which will cooperate with the existing ones and interact with them by means of a standardized protocol, e.g. the Agent Communication Language (ACL) [6]. This differs from the classic multi-tier architectures in which the relationships between the components at a particular tier are much stronger. Conceptually-wise, we define different layers in the system architecture in order to present more systematically the functionality of the middleware that is being developed. Implementation-wise, the middleware architecture is considered as a set of interacting intelligent agents.

Communication between a user mobile devices and a serving InfoStation could be realized in two ways:

- The InfoStation takes the initiative (the InfoStation's agents are pro-active), or

- The mobile device takes the initiative (the Personal Assistant agent is pro-active).

In the current prototype implementation of the middleware, the former approach is used for Bluetooth communication, whereas the latter applies for WiFi communication. The rest of this section presents the WiFi-based version of the middleware (Figure 1). For its development we chose to use the Java Agent DEvelopment (JADE) framework [7]. The middleware architecture consists of two parts [8]:

- **Core –** contains agents which ensure context-awareness and adaptability of the architecture. These agents are able to detect and identify different events in the environment, and correspondingly to generate dynamically other agents, which to serve/process the execution of the user's service request. The core is application independent and is deployed fully on each InfoStation.
- **Dynamic part** – contains all agents that are dynamically generated in response to the identified events (changes in the environment). These agents are responsible for the execution of scenarios and controlling the user service sessions. After completing their task, these agents are removed from the system architecture.

The core contains three fundamental agents – Agent Management System (AMS) agent, Reception Agent (RA), and Service Manager Agent (SMA) [9]. The AMS agent (a system agent for the JADE platform) controls the core's integrity. It identifies all events related to the creation, existence, and removal of agents in the architecture. It also allows agents to 'subscribe' for 'interesting' events. When a particular event takes place, the AMS sends notification messages to all relevant subscribers. There is only one AMS agent on each InfoStation. The RA supports a registry of all personal assistants currently within the range of a particular InfoStation. The registry holds records such as personal assistant's name, created/removed attribute, time of event, etc. To achieve its task, this agent subscribes with the AMS for 'interesting' events such as the appearance/disappearance of personal assistants. The SMA supports a catalogue of all services accessible locally on a particular InfoStation (i.e. the LmS services deployed on this InfoStation). For each offered service, the catalogue keeps records of the service profile (identification, type, parameters) and the profile of the servicing interface agent.

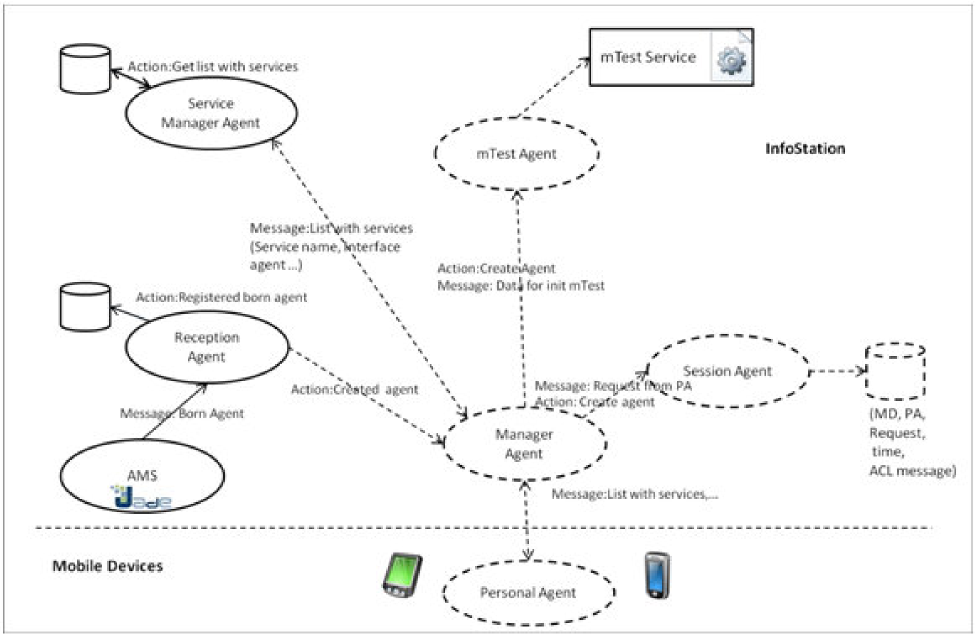

Fig.1. The WiFi-based middleware InfoStation’s architecture

The existence of dynamically generated agents depends on the identified situation in the environment such as the establishment of initial communication and maintaining the interaction between the mobile device and the InfoStation, control of user service sessions, service execution, etc.

For facilitating the establishment of initial communication between the mobile device and the InfoStation, there are two components (for simplicity not shown in Figure 2) working at the communication layer - an IS-server on the InfoStation and an IS-client on the mobile device. When a WiFi-enabled mobile device enters the service area of the InfoStation, the IS-client notifies the IS-server about its existence, establishes a connection with the IS-server, and confirms its readiness for exchange of data. This initial communication is needed in order to activate the Personal Assistant (PA) agent deployed on the mobile device. The subsequent interaction between the mobile device and the InfoStation is possible after registration (in AMS) of the PA agent. The AMS informs the RA about the presence of a new agent in the architecture (by sending this agent’s ID). The RA registers the new PA and generates a Manager Agent (MA) for serving the newly registered PA. Starting from this moment, the MA is the only agent that the PA can communicate with during the service request processing.

The user’s service session is controlled by the MA. For this, the MA generates its own helper called a Session Agent (SA). The SA registers all events caused by the user during his/her session and stores all relevant data (coming from the MA) such as the IDs of the mobile device and its PA, requests, time of occurrence, ACL messages, etc. Next the MA sends a message to the SMA requesting the list of services that this particular PA is authorized to access on this InfoStation. When the list is delivered back, the MA forwards it on to the PA, which facilitates its visualization on the screen of the mobile device. Next the user selects for execution some services from the list and this request is sent to the MA (via the PA). The MA splits this coming information in two parts. The first part, related to the service session, is sent to the SA for registration of the user action. The second one, related to the service request itself, is analyzed by the MA. After identifying the type of the service, the MA takes the necessary actions for satisfying the service request. More specifically, it generates an interface agent and passes to it the relevant data needed for the initialization of the service. All information exchanged between the user and the service is communicated on the established 'PA - MA - interface agent' channel.

3. Simulation environment

In order to test the middleware, we are developing simulation environment called CA^{3}SiEnv. It will enable us to:

- Find and correct errors more efficiently;
- Examine the behaviors of separate agents and the middleware as a whole;
- Optimize the middleware;
- Simulate execution of the scenarios in the IS architecture;
- Present visually the processes in the middleware and their tracing on three levels – scenario, agent and behavior levels.

In order to be fully integrated into the middleware, the simulation environment is agent oriented based on JADE. The core of CA^{3}SiEnv is Control Center module, which manages the components of the simulation system.

The main components of the CA^{3}SiEnv are Control Center (CC) and simulated InfoStation (sIS). Control center is the core of the simulation system. It is one JADE platform with agents that control the behavior of simulated environment and its events. CC hosts two levels of agents: Simulation Controller and sIS handlers.

Simulation Controller (SC) is the main agent in the simulation environment. It controls all of the aspects of the simulation. It provides GUI to the user, which represents the state of all simulation elements. Its role is to interpret user commands, translate and break them down to manageable sub-commands and relay them to sIS handlers. It also visualizes response from the handlers.

sIS hadlers are agents that manage the communication with simulated InfoStations. For every sIS created in the environment, there is one sIS Handler. Its role is to interpret commands from SC agent, if necessary break them to sub-commands and execute them. The execution of these commands may involve creating, destroying, sending commands to or receiving commands from simulated InfoStation. sIS handler communicates with dedicate agent inside sIS – the Spy agent. It visualizes changes in the simulated IS, which are messaged by the Spy.

Simulated InfoStations are platforms that contain standard CA^3 middleware (set of unmodified agents) with additional agents. These additional agents perform control and monitoring functions – they emulate changes in the real environment and detect the resulting changes in the middleware. These functions are delegated to the Spy agent.

Spy agent is situated in every sIS, along with standard CA^3 middleware agents. It communicates with its handler inside the ControlCenter platform, receiving commands and sending updates of status of the middleware inside sIS platform. Reacting to different commands, it causes changes of the environment of the middleware, as if they were caused by external influence. It also monitors changes of the middleware, such as creating and destroying agents from the dynamic part of the architecture, registration and deregistration of services, message exchange, changes in agent behaviors, etc. Information for these changes is sent back to the respective sIS handler, which is responsible for the visualization.

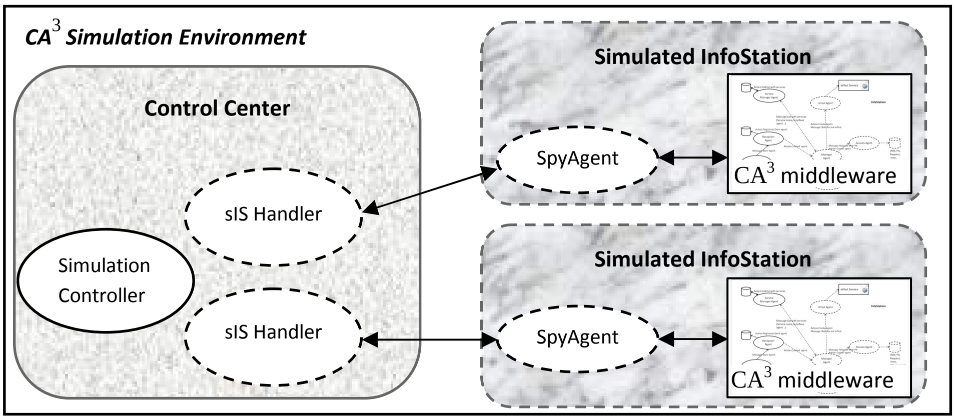

Fig. 2 CA^3 Simulation Environment

For the presentation of the events in different simulated IS platforms we are developing several levels of visualization. The main levels are:

- scenario level;
- agent level;
- behavior level.

Each level presents events from the point of view of the main element of the level, namely: scenarios, agents and behaviors.

In the scenario level will be displayed main tiers of InfoStation Architecture – Mobile devices, InfoStations and InfoStaionCenter. This level will follow up the execution of various scenarios.

Agent level will follow the events and changes occurring in middleware during the execution of scenarios.

The behavior level will monitor ongoing changes in the structure of agents during the execution of scenarios.

4. Conclusion

The proposed version of the middleware provides context-awareness and adaptability, regardless of the specific proposal. We are looking for opportunities to extend the middleware, so as to maintain application dependent context-awareness and adaptability. For our eLearning application this is support for students', learning discipline (domain) and pedagogical models.

5. Acknowledgment

This work was supported by the Bulgarian Science Fund under Grant No. Д002-149/16.12.2008.

References

[1] R. H. Frenkiel and T. Imielinski, “InfoStations: The joy of ‘many-time, many-where’ Communications”, WINLAB, Technical Report TR-119, April 1996.

[2] S. Stoyanov, I. Ganchev, M. O’Droma, D. Mitev, I. Minov, Multi-Agent Architecture for Context-Aware mLearning Provision via InfoStations, The 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, pp.549-552, October 28-31, 2008, Paris.

[3] Stojanov, S, I. Ganchev, M. O’Droma, E. Dojchev, G. Cholokov, Agent-oriented Distributed eLearning Center, Proceedings of the 2th Balcan Conference in Informatics BCI2005, 17-19 November 2005, Ohrid, Macedonia, pp. 238-245.

[4] S. Stoyanov, Context-Aware and Adaptable Architecture CA^3, REMIA’2010, 10-12.December 2010, University of Plovdiv.

[5] Stoyanov S., I. Ganchev, I. Popchev, M. O'Droma, An Approach for the Development of a Context-Aware and Adaptive eLearning Middleware**,** In: Intelligent Systems: From Theory to Practice, SCI 299. V. Sgurev et al. (Eds.). Springer-Verlag Berlin Heidelberg, 2010, ISBN: 978-3-642-13427-2. DOI: 10.1007/978-3-642-13428-9_26. Pp. 519-535.

[6] FIPA Agent Communication Language.
http://www.fipa.org/repository/aclspecs.html

[7] F. Bellifemine, G. Caire, and D. Greenwood, "Developing Multi-Agent Systems with JADE," John Wiley & Sons, Ltd, 2007.

[8] I. Ganchev, S. Stoyanov, M. O'Droma, V. Valkanova, D. Meere, Pervasive InfoStation-based mLearning System, ICSNC 2010: The Fifth International Conference on Systems and Networks Communications, August 22-27, 2010 - Nice, France, 320-325

[9] S. Stoyanov, I. Ganchev, I.Popchev, M.O'Droma, V. Valkanova, Agent-Oriented Middleware for InfoStation-based mLearning Intelligent Systems, 5th IEEE International Conference on Intelligent Systems IS'10, 07.07. – 09.07**.**2010, London, IEEE Catalog Number: CFP10802-CDR, ISBN:978-1-4244-5164-7, Library of Congress:2009934065, pp. 91-95

Ivan Minov,
Faculty of Mathematics and Informatics
University of Plovdiv
236 Bulgaria Blvd.
4003 Plovdiv, Bulgaria
e-mail: iminov@uni-plovdiv.bg

Damyan Mitev,
Faculty of Mathematics and Informatics
University of Plovdiv,
236 Bulgaria Blvd.
4003 Plovdiv, Bulgaria
e-mail: damyanmitev@uni-plovdiv.bg