Implementation and Deployment of an Intelligent Framework for Utilization within an InfoStation Environment

Zhanlin Ji
BioEngineering & Sensor Technology Research Center, University of Science & Technology Beijing, China
Email: zhanlin.ji@ieee.org

Damien Meere, Ivan Ganchev, Máirtín O'Dróma
Telecommunications Research Centre, University of Limerick, Ireland
Email: {Damien.Meere, Ivan.Ganchev, Mairtin.ODroma}@ul.ie

***Abstract*—With the continuing evolution of modern handheld mobile devices, an opportunity has arisen for a more comprehensive integration of these devices into educational environments. This paper details the design and implementation of an intelligent, light-weight, distributed Java platform based framework for the provision of mobile services (mServices) within an InfoStation-based mLearning environment. A multi-agent system, acting as a conduit between the mService applications and the mobile users is detailed. The deployment of this system for the integration of the supported services into a blended learning experience for system users is also outlined.**

***Index Terms*—mService; mLearning; InfoStation; Framework; Java; JADE; Drools**

I. Introduction

Throughout the last decade, mobile phones have continued to evolve and develop, to the point where now, these devices can offer functionality comparable to that of desktop computers. In recent years there have been strides made towards the utilization of these devices within the educational domain. Educational institutions are presented with an opportunity to enhance their existing educational infrastructures through the incorporation of mobile technologies, in order to provide a pervasive blended learning experience. A mobile eLearning (mLearning) system, designed to facilitate this bended learning experience has been presented in [1-4], as an 'anytime-anywhere' oriented system, utilized for the delivery of multimedia service content to mobile devices (cellular phones, PDAs, laptops, ultra-mobile PCs - UMPCs, tablets, etc) [5]. However, the multimedia content is personalized and customized to suit the accessing mobile devices and sent to mobile user through the best wireless connection available under the always best connected and best served (ABC&S) communication paradigm [5]. The mLearning system contains a context-aware and agent-oriented middleware built upon an InfoStation-based network architecture (Figure 1), consisting of the following components:

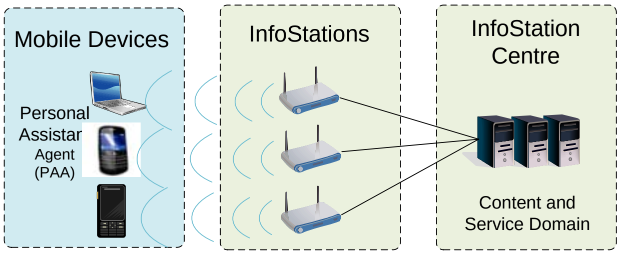

Figure 1. The InfoStation-based network architecture.

1) *Mobile devices* with installed personal assistant agents (PAAs). Running on a separate device, each PAA facilitates the initial authentication, authorization and accounting (AAA) of the user, personalizes the user requests for mobile services (mServices), i.e., mTest, mLecture, mLibrary, etc., and filters out and visualizes the results of the mServices execution.

2) *InfoStations* which facilitate mobile users with access to the mServices via the "best" available wireless connections.

3) *InfoStation Center* that includes: an InfoStation manager, which controls, updates and synchronizes information across all connected InfoStations within the network; user profile managers, which record mobile user personal information, including user location, ID, test results, interests, mobile devices, etc; mService managers, which facilitate the delivery of personalized mServices content to users based on their profiles.

Ideas pertaining to the concepts of InfoStation-based network architectures have been discussed previously in [1-4]. In [1], the implementation of a context-sensitive InfoStation-based multi-agent system (MAS) facilitating mServices was presented. A Java Agent DEvelopment framework (JADE) based MAS in conjunction with composite capabilities / preference profiles (CC/PP) was utilized to facilitate mServices personalization and contextualization. In [2], four main communications scenarios specific for this environment were presented: namely "no change", "change of mobile device", "change of InfoStation" and "simultaneous change of mobile device and InfoStation" scenarios. In [3], a description of

doi:10.4304/jsw.7.5.935-942

the main mServices, along with sample interactions were proposed. In [4], concepts relating to the control and management of service sessions was discussed, detailing a more sophisticated mServices presentation through the utilization of the Ontology Web Language for Services (OWL-S) protocol. Various associated implementation issues were also outlined in that paper.

Differing from the aforementioned research, within this paper, the design and implementation of a light-weight intelligent framework for the delivery of mService applications is detailed. Each application runs within a single container and utilizes agents to facilitate communications between each other. It is a basic framework which enables this mLearning system to run in a light-weight, distributed and intelligent way. Being defined as a tiered distributed system, the designed mLearning system has the following benefits:

1) *Scalability and extensibility* both horizontally (i.e. adding a new business rule within any tier does not require any alteration in the tier) and vertically (e.g. the InfoStation tier can be migrated or updated easily).

2) *Configurability, maintainability and security*: Maintaining the system's configuration parameters and security roles in the system is quite simple. Different tiers may define different rules for end users with different levels of security policy.

3) *Tiered distribution*: The mServices can be implemented /deployed on different machines and areas with different IP addresses. This approach is useful for implementing a level of localization of services within different geographical areas, for example within various faculties throughout a University campus.

The rest of the paper is organized as follows. Section II outlines some existing frameworks, upon which this system has drawn from. Section III details the design and implementation of the various aspects of this novel framework. Section IV details the incorporation of context awareness within the system. Section V discusses the deployment of this system and its incorporation into a blended learning environment. Finally Section VI concludes the paper.

II. Existing Frameworks

Java has proven to be the ideal technology for the development of this mLeaning system, due, in no small part, to its provision of solutions for a wide range of networks, computers and devices. On account of its versatility, efficiency, portability, and security properties, over 3 billion mobile phones have been built incorporating Java. There are also more than 6.5 million Java developers who are active within the Java dedicated community [6-7]. Java technology is structured into four editions: Java Enterprise Edition (Java EE), Java Standard Edition (Java SE), Java Micro Edition (Java ME), and Java Card. There are several frameworks available for development of mLearning systems within a Java environment: Spring [8], Mobile Information Device Profile (MIDP) applications [9], Google Android [10] and Google Web Toolkit (GWT) [11], Drools [12], and the Java Agent DEvelopment Framework (JADE) [13]. Essentially, Java can be thought of as the lowest common denominator for catering to the largest possible audience. While a great deal of mobile phone users have adopted the modern ‘Smart' phone, a great many people still utilize more basic devices. By building this system on top of various Java-based technologies, the services offered can be delivered to everyone fairly.

Mobile Information Device Profile (MIDP)-based applications run on the top of the Java ME Connected Limited Device Configuration (CLDC) profile. MIDP, in conjunction with the CLDC, provides a Java runtime environment for a multitude of mobile devices. Requiring only 128kB of memory, MIDP is ideal for the development of low-end mobile applications. MIDP provides UI APIs, input- and event handling mechanisms, persistent storage solutions, network connectivity, etc.

Google Android is an open-source software platform, which incorporates a small Linux operating system, a middleware and a number of associated APIs. It has become part of the Open Handset Alliance (OHA) and is suited to the development of PDA-based mobile applications. Asynchronous JavaScript and XML (AJAX) are widely used for creating client-side web applications or rich Java applications, providing a better user experience. GWT is suited for UMPC/PC-based AJAX applications.

Spring is an open-source software framework within which to implement Java applications. This framework is widely utilized both for web-based thin applications and indeed user interface (UI)-based thick applications. It is the most popular framework in the Java community, first and foremost on account of its simplicity. In comparison with Java EE enterprise Java bean (EJB), Spring-based applications do not require an EJB container thus reducing the system complexity. The Spring framework also greatly aids the testing process, by supporting Java classes for unit-testing and integration-testing. This latter feature is suited for test-driven development and inherent portability, as the Spring-based application can run within low-end Servlet container environments, i.e., Apache Tomcat, Sun GlassFish, etc [8]. Also, the framework provides a number of useful services, for example Inversion of Control (IoC), Aspect-oriented Programming (AOP), Relation Database access with object-relational mapping tools, model-view-controller (MVC) model, authentication, etc. All of these services are configurable, customizable, and extensible, thus facilitating the development of Java EE applications with a level of simplicity and testability. The size of Spring Java's full library of APIs is about 3MB. Besides considering the issue of portability, the Spring framework is quite suitable for developing mService applications in mLearning systems.

JADE is a FIPA-specification compatible, open-source software framework, fully implemented in Java. It is suitable to act as a MAS middleware in our mLeaning system, facilitating the system's distributed ability. JADE consists mainly of three parts: a runtime environment supporting the execution of agents, a library for agent development, and a graphical tool facilitating the monitoring of agent operation within the environment. In JADE, each agent lives within a container, which can dynamically discover other agents for cooperation and communication in peer-to-peer mode. Currently, JADE lacks on uniform support for different mobile operating systems, as such, a lightweight re-design is required for this mLearning system.

Drools is an open-source, pure Java, rule engine platform, developed by JBOSS and enhanced by implementing the Rete algorithm for object-oriented systems with the forward chaining excitation method. Drools is ideally suited to act as a rule engine API within this mLearning system, charged with controlling the business's behaviours, dynamically changing the business logic, and efficiently managing the system. There are four components which constitute Drools: facts asserted into the working memory, rules stored in the production memory, pattern matchers used for deciding which rules are true for facts assertion, and agenda used to manage the execution order by a conflict resolution strategy. To effectively integrate this mLearning system with the rules engine, drools needs to be refactored to integrate it into the JADE system.

MIDP, Android and GWT applications act as client-side applications, running onboard the user's mobile device. These applications, in conjunction with the onboard PAA, communicate with the agents operating within the InfoStation tier. Within the InfoStation Centre tier, Drools and JADE must be integrated into the Spring framework. However to date, few studies have taken place on the integration of Drools and JADE within the Spring framework. In order to build an intelligent light-weight (small foot-print, less than 200 kilobyte) framework, this paper proposes a new light-weight Spring-like framework which integrates Drools and JADE under the IoC design pattern. A new design for JADE agents operating on mobile devices is also proposed.

III. Design and Implementation of the Intelligent Framework

The MAS environment, which constitutes the InfoStations, can be split into three separate layers: an agent's layer, an agent's support layer, and a services layer [1], as illustrated in Figure 2. The following is a short description of the interactions between the various agents within the InfoStation environment in facilitation an mService in the ABC&S mode. In order to facilitate the users with access to the various services housed within a particular InfoStation, a connection must first be established. This process is initiated by the Scanner agent, which continuously searches for mobile devices / PAAs within the service area of the InfoStation. It is this agent which initializes communication between the PAA and the agents in the InfoStation MAS. On receipt of a call-for-proposal (CFP) ACL message, this agent passes the message on to the Connection Advisor agent, and extracts information from the “UserDefinedHeaders”, regarding the context of the requesting device. This information can be passed on to the Scenario Manager agent whose role is to monitor and manage changes to the operating environment within which the services will be delivered (i.e. change in access device). In the event of a significant change of service environment, this agent gathers the new capability and preference information (CPI) via the Scanner agent. Then, in conjunction with the Query Manager agent and the Content Adaptation agent, it facilitates the dynamic adaptation of the service content to meet the new service context.

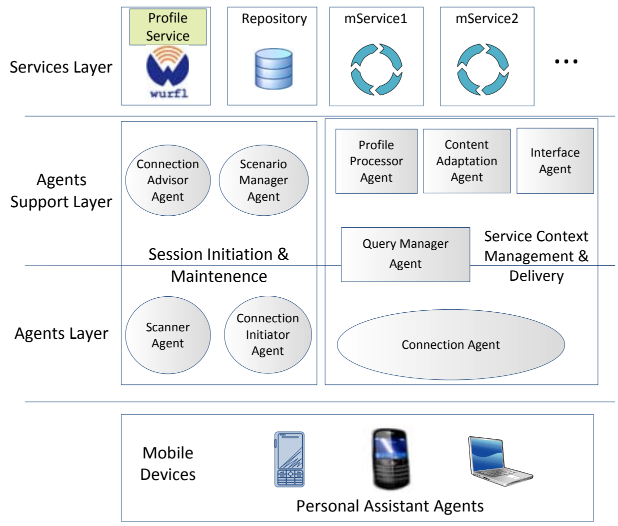

Figure 2. The agent-oriented InfoStation layered model.

The role of the Connection Advisor Agent (CAA) is to gather a list of services applicable to the user. The information required to generate this list is garnered from the headers contained within the received service request (CFP). Information needed for the filtration is stored in local DB4O databases. The CAA gathers a list of the classes, based on the course and year of study of the user, as specified in the ACL request header. Then the CAA examines the user credentials, querying them against a DB4O database of users. If the user's credentials pass these AAA procedures, the CAA gathers a list of services, applicable to the user, based on courses they study. Once a list of applicable services has been collated, the CAA passes on the filtered service list to the Connection Initiator agent, who takes on the task of initiating a connection with the PAA onboard the mobile device. Once this connection has been established via Bluetooth or Wi-Fi, the Connection Initiator generates an agent to which it hands over the control of the connection, called a Connection Agent. From this point on, all communications between the InfoStation and the PAA are directed by the Connection Agent.

The Query Manager performs the role of an intermediary between the agent layer and the agent support layer. It determines where information received from the mobile device is to be directed, e.g. directly to simple services, or via Interface Agents to sophisticated services. It also transforms messages coming from the Connection Agent into messages of the correct protocols to be understood by the relevant services, i.e. for simple services - UDDI or SOAP, or for increasingly sophisticated services by using more complicated, semantic-oriented protocols. The Query Manager agent also interacts with the Content Adaptation agent in order to facilitate the PAA with increasingly contextualized service content. This latter agent essentially performs the role of an adaptation engine, which takes in the profile information provided by the Profile Processor agent, and executes the requisite adaptation operations on the service content. The tasks undertaken by the Content Adaptation agent, the Scenario Manager agent and the Profile Processor agent enable the system to dynamically adapt to changing service environments, even during a particular service session. Once a particular service session has been initialized and the service content adapted to the requisite format, the Connection agent facilitates the transfer of information back to the user's mobile device.

To implement the aforementioned mService applications, an intelligent light-weight Spring-like framework was designed, by utilizing a light-weight HTTP container to facilitate an environment for running the aforementioned mLearning applications and processing the HTTP requests and responses. The framework is implemented purely in Java and as such is not dependent on any Java Archive (JAR) files. The high-level view of the architecture is depicted in Figure 3.

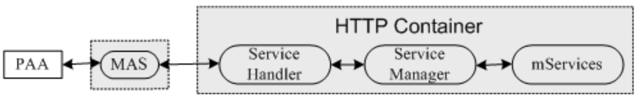

Figure 3. High-level view of the architecture from the implementation viewpoint.

A. HTTP Container Implementation

The light-weight HTTP container listens for incoming TCP connections from agents and processes the requests with the corresponding service (or handler). To achieve a light-weight design, the Sun pure Java httpserver package was utilized for the implementation of the HTTP server. After the InfoStation manager sends the start-up request to the HTTP container, the Boot service initializes the context for building a singleton factory to support the IoC container in the light-weight framework, starts up the JADE MAS, and boots up the HTTP server.

When the HTTP server is ready, a number of services, which act as HTTP handlers, are created for processing the requests and responses. The *createContext* function is the key element in the implementation of this HTTP container. It is used for mapping the aforementioned handlers to their respective services in the container. All handler objects are defined within the configuration file httpserver.ini. Each line in this file contains a URI path, such as "/mtest,mtestService", which represents the mTest service handler. The UML diagram of the light-weight HTTP container's main Java classes is shown in Figure 4.

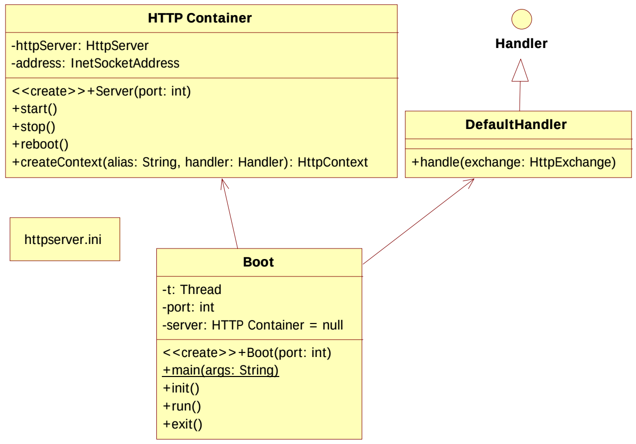

Figure 4. The UML diagram of the HTTP container's main classes.

B. Framework Implementation

Like Spring, the mLearning framework follows interface-based programming and IoC design patterns, and consists of the following tiers:

1) *Presentation tier*, which represents the HTTP service handler and encapsulates the required presentation logic;

2) *Business tier*, which provides the business services interfaces and consists of business logic for managing the services;

3) *Integration tier*, which is responsible for the actual algorithms execution, such as the model object construction, database operations, mService algorithm execution (integrated with Drools), etc.

Adhering to specified IoC design patterns, the HTTP handler need only retrieve a service from the factory, instead of having to create a new object each time. The tiered mLearning framework makes the system easier to maintain, manage and scale as well as looser coupling. Figure 5 illustrates the UML class diagram of the main interfaces and classes of the mLearning framework. Handlers receive HTTP requests from the MAS system, and forward the requests to the business logic tier for execution. Finally, the mService runs within the integration tier. The context class is the key design feature in the framework. All the integration tier's functions, i.e., database operations, mService algorithms, etc, are configured in the services.xml file with the following paired parameters - service name and corresponding Java class name. In the context class, the parameters are put into a service Map object. The HTTP container maintains this service Map. When the service logic runs an mService, the mService object can be gotten from the service factory instead of requiring the creation of a new Java object.

C. JADE Implementation

As the mLearning client application does not require direct access to the local resources, a hardware-independent uniform JADE design has been elaborated based on JADE-LEAP [14]. It can work on the mLearning client side using the same JAR library and does not require a MANIFEST configuration file. To achieve a light-weight architecture of the mLearning client, the system operates only in split-container mode, i.e., the JADE container is segmented into a front-end (running on the mobile device) and a back-end (running in the server side). To unify the mLearning mobile client design, a HTTP-like request/response mechanism (Figure 6) together with a compact Java object serialization APIs has been developed.

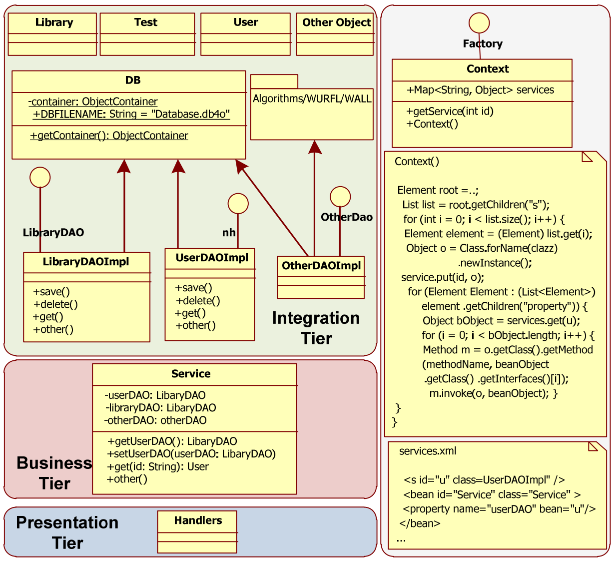

Figure 5. The UML diagram of the mLearning framework's main classes.

The client PAA consists of multiple generic behaviours. Two novel interface components have been designed (Figure 6): (i) a *IResultListener* operating on result (even though declared as an agent, it is implemented in the corresponding user interface (UI) in the mobile device); (ii) a *MessageListener* operating on handler (like emph*IResultListener*, it is registered and declared in the *MessageManager* but implemented in the corresponding UI). This design pattern

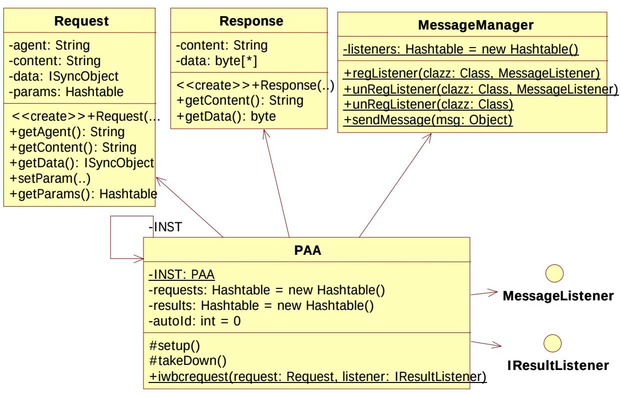

Figure 6. The HTTP-like request/response mechanism for JADE.

D. Drools Implementation

A rule-based expert system operates in the integration tier for separating the business logic from the mService application. As stated in Section II, Drools has been selected as the rule engine within this mLeaning system. With this interface design pattern, when a new rule is applied to the system, the system does not need to be recompiled, thus ensuring loose-coupling. Figure 7 shows the UML diagram of the Drools main recommendation classes. The RecomendCmd class implements the interface recommendation and gets the recommended Objects via a sort interface. To ensure the Algorithms_Drools has only one instance, The Singleton Design Pattern is used for this API implementation.

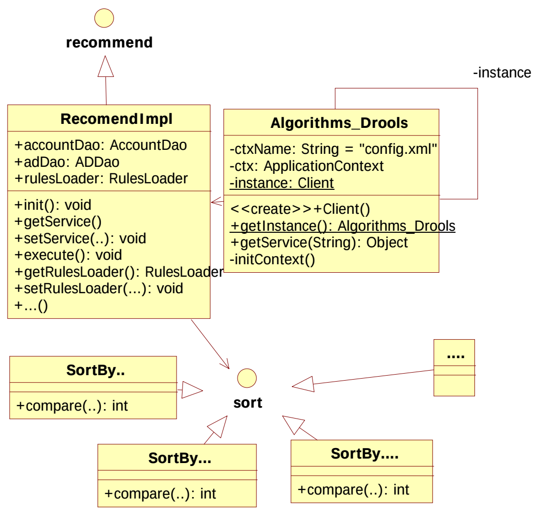

Figure 7. The UML diagram of Drools main recommendation classes.

E. Comparison

Compared with the Spring framework, this light-weight framework does not support the following techniques: Servlet, Java server page (JSP), MVC, AOP, and cannot run within the existing HTTP containers such as Tomcat, GlassFish, etc. But the chosen approach has the benefit of being independent on any JAR file, and the foot-print is only 200 kilobyte. It is particularly suited to resource limited mobile application design. This is the key feature of this light-weight design of the framework. Also, through the integration of JADE and Drools into the framework, the system can work in a distributed and far more intelligent manner.

IV. Context Awareness

One of the fundamental aspects of this system is the facilitation for service context-sensitivity and personalization. Due to the heterogeneous nature of the environment within which services are delivered, i.e. a huge variation in capabilities of user devices and wireless access types, it is necessary for service content to be adapted to meet possible constraints. However, within an educational domain, it is also necessary to take the context of the user into account when adapting the presented services. To facilitate this personalization and contextualization, the capability and preference information (CPI) must flow between the mobile devices and service providers, i.e. between the PAA and the InfoStation / InfoStation Centre. The agents operating within the JADE-based MAS, communicate information, such as user location, device details, profiles, service request details etc., via the Agent Communication Language (ACL) [13]. This information is utilized by the agents operating within the InfoStation to perform the AAA procedure and provide lists of services applicable to the authenticated user. It is imperative that the content, delivered to the user device, be delivered in a format suitable to that particular device. As such, the information gathered by PAA is enough to generate a user-Agent string, from which the capabilities of the accessing device can be ascertained. In order to facilitate this functionality, the Wireless Universal Resource File (WURFL) [15] was incorporated into the system. This XML configuration file, contains information about capabilities and features of a vast variety of mobile devices, and with its associated API, enables an agent within the InfoStation to utilize the generated user-Agent string, and to compile a directory of the capabilities and features of any device that the system may encounter. Of particular interest is the supported mark-up, which identifies the preferred content format of that particular device. Also incorporated into the InfoStation system, is the Wireless Abstraction Library by Luca (Passani)- Next Generation (WNG) [16], which enables the creation of service content in one mark-up and facilitates the rendering of this content as in WML, XHTML MP or CHTML automatically. Provided only with information about the device and the service chosen by the user, the InfoStation is able to dynamically render the content that suits the device. Not only does this benefit the users, as the content they receive is dynamically tailored to suit them and their devices, but the content developers need only generate their content in one format, and the system can adapt it accordingly to each service request.

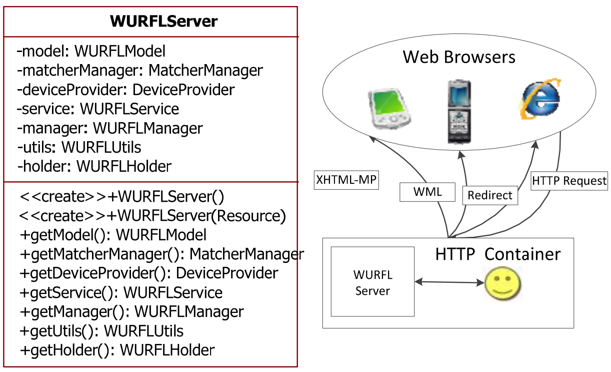

Figure 8. The WUFRLServer and WURFL Operational Diagram

In WALL design, for a Java version, the system requires a JSP/Servlet container (i.e., tomcat) for tag generation. This leads a pure Java re-development to let WALL function within the lightweight framework discussed in the previous section. A *WUFRLServer* was developed for determining the MT's capabilities and processing on the requested the response (Figure 8).

V. MLECTURE DEPLOYMENT

There have been many studies advocating the use of technology in education [17-20]. Typically, higher education institutions have been quite resistant to change, yet more and more institutions have begun to adopt and promote the use of blended learning and technology enhanced learning. [21] notes that 'gone are the days when university classes contained only highly selected students, with present day classes now containing students with a more diverse range of academic skills, past teaching and learning experiences, prior knowledge, approaches to learning and expectations of the tertiary experience'. This diversity and wide range of mixed-ability students leads to further challenges and issues for third-level educators. A diverse group of students calls for a re-evaluation of the teaching and learning strategies traditionally utilised within the third-level classroom. The traditional didactic 'jug and mug' model of teaching and learning is no longer appropriate for many third-level classrooms, as a wide variety of mixed ability students will enter the lecture hall with their own prior knowledge, experiences and learning styles. The blended learning approach can offer these students a more holistic experience of third-level education. The blended learning approach, rather than a total online learning experience, avoids the drawbacks of complete online learning and the lack of true social interaction it provides, which is a key component of the learning experience for many students [19]. However, while studies on the use of blended learning and online learning advocate these methods, they also call for the recognition of a number of pitfalls. In order for the students to fully engage with the learning experience, it is vital that the focus is on the material being taught, rather than on the technology. This calls for a piece of technology that can be seamlessly integrated in the students' lives and the lessons. This system seeks to harness the communicative potential of the students' own mobile devices in order to enhance their learning experiences. In recent years, various technologies have been embraced by the educational community, in particular the incorporation of media such as podcasting, or indeed the use of online collaboration and learning environments such as Moodle and Sakai. However, in the case of actual lecture delivery, these technologies have often been used to supplement teaching after the fact. The system discussed in this paper seeks to enable the utilization of mobile technology to enhance the delivery of educational content to students. In [22], Kinsella describes the utilization of mobile phones to interact with large classes over SMS. However, within our system, an enhanced mLecture service can be utilized to harvest feedback from large classes, without the cost usually associated with SMS. Figure 9 shows the mLecture deployment diagram. By utilizing an InfoStation, positioned within the lecture theatre, the class can be presented with questions, which the students can answer. The responses can then be sent, via PAA installed on their mobile devices, to the InfoStation. By harnessing this gathered information, the lecturer can gauge the student's assimilation of the presented material, and if necessary, dynamically alter the direction of the class to ensure an optimal learning experience. This process would also encourage the continued attendance of the

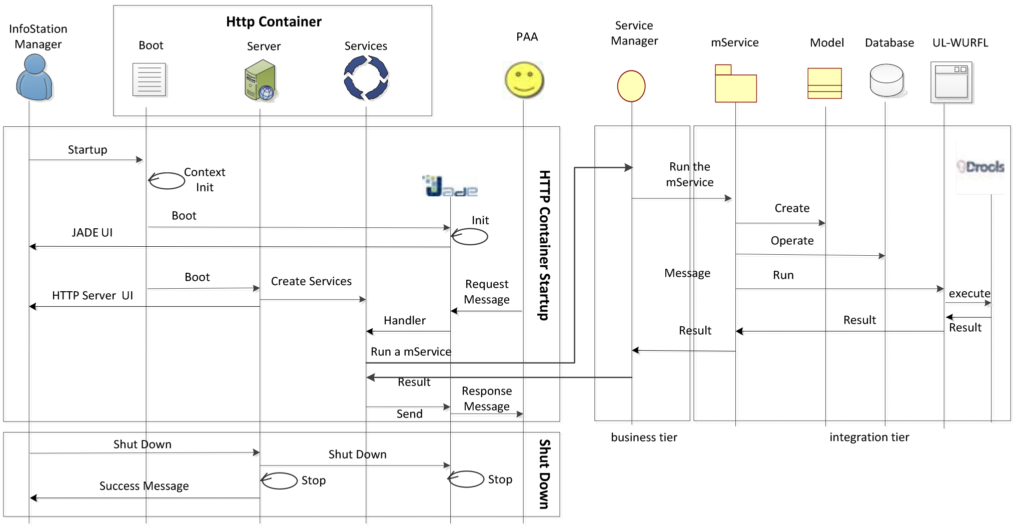

Figure 9. The mLecture deployment diagram.

students, enabling the lecturer to measure attendances during the lectures, as each PAA is provides details of the user associated with it. Of course the main use of this mLecture service is facilitating the users with “anytime-anywhere” access to the lecture content, aiding more continuous learning experience. Through the incorporation of this mLecture service, and for example the mTest service [4], the students can interact within a true blended learning environment.

VI. Conclusion

Various implementation aspects of an intelligent light-weight framework for use in mLearning systems have been presented in this paper. The framework utilizes pure Java language and as such is not dependent on any JAR files. A light-weight HTTP container was set out for the provision of an environment for running the framework-based mService applications. The framework consists of three tiers: a presentation tier, a business tier and an integration tier. The HTTP handler runs in the presentation tier, sending requests to the business tier and then executing the corresponding mService in the integration tier. An inversion of control (IoC) design method was proposed to make the architecture easier to maintain, manage and scale, as well as ensuring its loose coupling. The proposed framework has a multi-agent structure and is implemented by means of a modified Java Agent DEvelopment Framework (JADE) environment. With the JADE module, the framework can operate in a distributed way. Drools was selected as a rule-based expert system integrated into the integration tier for separating the business logic from the mService application. WURFL and WNG have been incorporated into the system to enable it to adapt to the heterogeneous nature of the target devices, providing the service content in the most suitable format. The deployment of this system within a lecture theatre environment for facilitating a blended learning experience was outlined, highlighting the necessity for a departure from the traditional teaching and learning strategies, and identifying some of the benefits this system can offer to a more holistic learning experience.

Acknowledgment

This publication has been supported by the Irish Research Council for Science, Engineering and Technology (IRCSET) and the Telecommunications Research Centre, University of Limerick, Ireland.

References

[1] D. Meere, *et al.*, "Adaptation for Assimilation: The Role of Adaptable M-Learning Services in the Modern Educational Paradigm," *International Journal "Information Technologies and Knowledge" (IJ ITK)*, vol. 3, pp. 101-110, 2009.

[2] I. Ganchev, *et al.*, "InfoStation-Based Adaptable Provision of m-Learning Services: Main Scenarios," *International Journal "Information Technologies and Knowledge"*, vol. 2, pp. 475-482, 2008.

[3] I. Ganchev, *et al.*, "An InfoStation-Based Multi-Agent System Supporting Intelligent Mobile Services Across a University Campus," *Journal of Computers*, vol. 2, pp. 21-33,, May 2007.

[4] S. Stojanov, *et al.*, "Semantic Multi-Agent mLearning System," in *Semantic Agent Systems: Foundations and Applications (SASFA 2010)*, A. Elci, *et al.*, Eds., ed Berlin Heidelberg: Springer-Verlag, 2010, pp. 243-272.

[5] M. O'Droma and I. Ganchev, "Toward a Ubiquitous Consumer Wireless World," *IEEE Wireless Communications*, vol. 14, pp. Pp. 52-63, February 2007.

[6] J. Gosling, *et al.*, *The Java(TM) Language Specification, (3rd Edition)*, 3 ed.: Addison-Wesley Professional., 2005.

[7] *Java.com, [Online]. Available - http://www.java.com/en/about/, [Accessed: June 22, 2011].*

[8] D. Kayal, *Pro Java EE Spring Patterns: Best Practices and Design Strategies Implementing Java EE Patterns with the Spring Framework*. Berkeley, CA, USA: Apress, 2008.

[9] L. Huan-huan, *et al.* (2009) Mobile development of Java ME based on eclipse. *Mechanical & Electrical Engineering Magazine.*

[10] J. DiMarzio, *Android: A Programmer’s Guide, 1st Edition.* New York, NY, USA.: McGraw-Hill, Inc., 2009.

[11] E. Burnette and A. Houghton, "Hands-on Google Web Toolkit.," *Dr Dobbs Journal,* vol. 33, pp. 26-30, 2008.

[12] M. Proctor. (2007, *Drools Documentation, Version 4.0.3,[Online]. Available - http://www.jboss.com/products/rules, [Accessed - June 22, 2011].*

[13] F. L. Bellifemine*, et al.*, *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*: John Wiley & Sons, 2007.

[14] A. Moreno*, et al.*, "Using JADE-LEAP to implement Agents in Mobile Devices," *EXP in search of innovation - Telecom Italia,* September 2003.

[15] WURFL. June 22). *Wireless Universal Resource File (WURFL) [URL] - http://wurfl.sourceforge.net/.*

[16] L. Passani. June 22). *Wall: Next Generation [URL] - http://wurfl.sourceforge.net/wng/.*

[17] C. H. Crouch and E. Mazur, "Peer Instruction: ten years of experience and results," *American Journal of Physics,* vol. 69, pp. 970 – 977, 2001.

[18] D. R. Garrison and H. Kanuka, "Blended learning: Uncovering its transformative potential in higher education," *The Internet and Higher Education,* vol. 7, pp. 95-105, 2004.

[19] A. Heinze and C. Proctor, "Reflections on the use of blended learning," presented at the Education in a Changing Environment, 2004.

[20] J. R. MacArthur and L. L. Jones, "A review of literature reports of clickers applicable to college chemistry classrooms," *Chemical Education Research and Practice,* vol. 9(3), pp. 187 – 195, 2008.

[21] P. Zeegers and L. Martin, "A Learning-to-learn Program in a First-year Chemistry Class. ," *Higher Education Research and Development.,* vol. 20, pp. 35-52, 2001.

[22] S. Kinsella, "Many to one: Using the mobile phone to interact with large classes," *British Journal of Educational Technology,* vol. 40, pp. 956-958, 2009.

Zhanlin Ji received his MEng, M.Sc and PhD degree from Dublin City University, Beijing University of Posts and Telecommunications, and University of Limerick in 2005, 2006 and 2010, respectively. Since 2010, he is a lecturer at the Hebei United University, China, a PostDoc Researcher in the University of Science and Technology Beijing, China, and a Research Fellow (External) of TRC, University of Limerick, Ireland. His research interests include UCWW, WBCs, Software Engineering, Mobile and Ubiquitous Computing, Information Retrieval and Management, Content awareness, Personalization, Multi-agents systems, Java EE architectures, Medical image analysis and processing, Machine Vision, Broadcasting.

Damien Meere received his BSc degree in IT&T from University of Limerick, Ireland, in 2005. He is currently pursuing his PhD within the Telecommunications Research Centre at University of Limerick. His research is focused on the provision of contextualised and personalised mobile eLearning (mLearning) services across the University campus area, primarily facilitated through the use of intelligent mobile Agents.

Ivan Ganchev is a Senior Member of the Institute of Electrical and Electronic Engineers (IEEE), USA. He received his engineering and doctoral degrees from the Saint-Petersburg State University of Telecommunications in 1989 and 1994, respectively. He is an Associate Professor at the University of Plovdiv and an ITU-T Invited Expert. Currently he is lecturing in the University of Limerick (Ireland), where he also acts as a Deputy Director of the Telecommunications Research Centre. Previously he served as a member of the Academic Network for Wireless Internet Research in Europe (ANWIRE) and two European Science Foundation 'COoperation in the field of Science and Technology research' Actions: 'Modelling and Simulation Tools for Research in Emerging Multi-service Telecommunications', and "Traffic and QoS Management in Wireless Multimedia Networks", (COST 285 & 290). Currently he is a member of the COST Actions IC0906 "Wireless Networking for Moving Objects" (WiNeMO) and IC0905 "Techno-Economic Regulatory Framework for Radio Spectrum Access for Cognitive Radio/Software Defined Radio" (TERRA). His research interests include: simulation and modeling of complex telecommunication systems, new communications paradigms for wireless next generation networks (NGN), always best connected & best served (ABC&S), third-party authentication, authorization and accounting (3P-AAA) management, wireless billboard channels (WBC) and cognitive pilot channels (CPC), Internet tomography, mLearning. Dr Ganchev has served on the Technical Program Committees of a number of prestigious international conferences and workshops.

Máirtín O'Dróma received his BE and PhD degrees from the National University of Ireland in 1973 and 1978, respectively. He is a Senior Academic and Director of the Telecommunications Research Centre at the University of Limerick, Ireland. He is an IEEE Subject Matter Expert, an ITU-T Invited Expert, a Fellow of the IET and Chair of Ireland's Royal Irish Academy's Communications and Radio Science Committee. His previous activities include: founding partner and steering committee member, TARGET -Top Amplifier Research Groups in a European Team-, the EU FP6 Network of Excellence IST-507893, 2004-2008, and section head of the RF power linearization and amplifier modeling research strand; founding partner and steering committee member, ANWIRE -Academic Network for Wireless Internet Research in Europe, the EU FP5 Thematic Network of Excellence IST-38835, 2002-2004; member of two European Science Foundation 'COoperation in the field of Science and Technology research' Actions: 'Modelling and Simulation Tools for Research in Emerging Multi-service Telecommunications', and "Traffic and QoS Management in Wireless Multimedia Networks", (COST 285 & 290). Currently he is delegate (Ireland) to the COST Action IC0906, "Wireless Networking for Moving Objects" (WiNeMO). His research interests include: wireless NGN infrastructural innovations and new paradigms; complex wireless telecommunication systems simulation and behavioral modeling, linearization & efficiency techniques in multimode, multicarrier broadband nonlinear microwave and mm-wave transmit power amplifiers; smart antenna and MIMO channels. Dr O'Droma served on the Technical Program Committee of numerous IEEE and other international conferences and workshops.