Development of InfoStation-based and Context-aware mLearning System Architectures

Ivan Ganchev*, Stanimir Stojanov**, Máirtín O'Droma* and Damien Meere*

**Telecommunications Research Center, University of Limerick, Limerick, Ireland.*

***Department of Computer Systems, Plovdiv University "Paisij Hilendarski", Plovdiv, Bulgaria.*

1. Abstract

This chapter¹ considers the main aspects in the development of InfoStation-based and context-aware multi-agent system architectures, which facilitate the provision of intelligent mobile eLearning (mLearning) services across a University Campus area.

Generic models and approaches for the development of eLearning systems are considered, paying particular attention to the various aspects which will have a greater bearing on the adaptation and personalization of the eLearning content for the specific end user. The supporting network architecture is described both horizontally and vertically illustrating how each of the main system components collaborates in order to facilitate the delivery of mLearning services. The chosen multi-agent approach for the system implementation is justified. The approach uses some fundamental OMG-MDA ideas with additional elements, which take into account the specifics of the InfoStation network. The software architecture required to provide the needed hardware flexibility and adaptability according to the requirements of modern eLearning systems is discussed. The system architecture is being developed on three levels: scenarios level, agent-oriented middleware, and eLearning services level. The four main generic service scenarios are presented highlighting the requirements for effective distribution of service control and session management between the agents. Suitable agent-oriented system models are proposed and explained, and the interactions between agents and mLearning services in each of these models are presented. Approaches for system implementation and structuring are also considered.

Particular attention is paid to the creation of user profiles and service profiles aiding the delivery of more personalised and context-aware mLearning services and their implementation through the Composite Capabilities/ Preference Profile (CC/PP) and User-Agent Profile (UAProf). The utilization of J2ME as the architecture for the development, deployment and execution of the mLearning services is outlined. The implementation of the system by means of the Java Agent DEvelopment (JADE) framework making particular use of its Light Extensible Agent Platform (LEAP) module is also discussed in detail.

¹ Authors wish to acknowledge the support of the Bulgarian Science Fund under Grant No. Д002-149/2008, the Irish Research Council for Science, Engineering and Technology (IRCSET), and the Telecommunications Research Centre, University of Limerick, Ireland (<http://www.ece.ul.ie/trc/>).

2. Introduction

The InfoStations paradigm is a new infrastructural system concept proposed to provide “many-time, many-where” (Frenkiel and Imielinski 1996) wireless data services. In this chapter we propose an InfoStation-based system architecture operating across a University Campus area allowing registered users to access a range of eLearning services by means of mobile devices (cellular phones, laptops, personal digital assistants- PDA’s). The InfoStations can essentially be thought of as portals through which the mobile devices communicate with specific eLearning services. This infrastructural system architecture provides an ideal opportunity to enhance the mobile eLearning (mLearning) experience and could be used as a platform for other supplementary communications services supporting the mLearning process. In addition it satisfies the requirements for highly adaptable eLearning architecture (Cook, Harrison et al. 2003) by providing:

- Improved ability to adapt rapidly and transparently to changes in learners’ profiles and their eLearning progress;
- Better mechanisms for discovering and comparing relevant learning resources;
- An ability to specify the requirements of an instructional component and delegate the discovery of a resource satisfying it to another process (e.g. a software agent);
- Easy interoperation with both external resources and other learning management systems (LMS).

The proposed system architecture has the following distinguishing features:

- New architecture type (service-oriented), allowing more flexibility during the run-time (by the use of Semantic Web and Semantic Grid);
- Open architecture allowing multiple reuse of eLearning content, e.g. by means of the ADL SCORM standard etc;
- Multi-communication access to learning resources by means of mobile devices;
- Agent-oriented approach;
- Knowledge technologies;
- Application integration;
- Dynamic profiling.

The objective of this chapter is to introduce the architecture of the InfoStation-based eLearning system, illustrating its flexibility and intelligence, all of which are enhanced through the utilization of intelligent agents, which communicate with the systems functional modules.

The organization of the chapter is as follows. In section 3 we consider generic models and approaches for development of eLearning systems. In section 4 we present the InfoStation-based network architecture and its functional entities. Section 5 describes the proposed multi-agent system structure and considers two possible agent-oriented system models. Sample mLearning services used in our system are considered in Section 6. Section 7 is dedicated to the implementation issues. Finally section 8 concludes the chapter.

3. E-Learning Systems: Models and Approaches for Development

While it is important to ensure that an eLearning system operates and functions efficiently from a technological point of view, we must also be sure to consider the system from a purely educational standpoint. In the past, the primary role of this type of technology in

education was to facilitate remote learners with the capabilities to learn as though they were campus-based. In this case, the enhancement was more pragmatic than pedagogic, enabling a greater access to learning rather than a greater understanding of a particular concept. However, in recent years the view of what constitutes eLearning and where its value lies has most certainly shifted. ELearning has become a cheaper, easier and far more accessible alternative to the traditional face-to-face learning paradigm. Of course while providing a viable alternative, it has also become an invaluable aid to accompany the traditional teaching approach and as such create a more blended learning environment (Ganchev, Stojanov et al. 2006). Essentially we must think of an eLearning system as the utilization of technology to achieve greater learning outcomes as well as a cost-efficient method for the delivery of the learning environment to the learners.

Biggs (Biggs 1999) describes the task of good pedagogical design as one of ensuring that there are absolutely no inconsistencies between the curriculum we teach, the teaching methods we use, the learning environment we choose, and the assessment procedures we adopt. Therefore in the design of this system, we must first carefully specify the learning outcomes we wish to achieve, and as such design various teaching activities which will enable the students to achieve these desired outcomes. Indeed the assessment tasks need to be designed so as to genuinely test whether or not a particular learning outcome has been achieved. Essentially, while the technology available to us continues to develop at an astounding rate, we must never lose sight of the educational aspects. Indeed by concentrating on using newer technologies to their greatest potential, we run the risk of drawing a student's attention away from the learning content and the educational goals we wish to achieve.

3.1 Conceptual Model

In general each eLearning system consists of five components (Figure 1) the main two of which are:

- *eContent generator* – used by educators and instructors to create an electronic content (eContent) needed for the learning process;
- *eContent interpreter* – used by learners for the interpretation of the created eContent.

Fig. 1. The Conceptual Model of an eLearning System

The creation and interpretation of the eContent depends on some important circumstances and limitations, which could be generalized in the following three (sub) models:

- *Domain model* – represents the knowledge of a particular learning area/field; usually contains basic (atomic) concepts and different semantic links between them;
- *Pedagogical model* - contains information about possible didactical solutions used by the learning management system (LMS), e.g. to explain how to study a particular eLearning material/content. The possibility for presentation and control of learning tasks plays an important role in this model;
- *User model* – usually stores specific information for individual users, which could be used for personalization of their eLearning process. Depending on the chosen approach this information could represent the preliminary degree of learners' preparation, preferences, affinity to a particular course/module, type of mobile devices in use, ID data etc. The following main approaches could be used for the implementation of this model:
 - Stereotypes – the simplest way of modeling by user categorization in groups (categories). For each group the system supports corresponding description called *stereotype*. Each user is categorized/assigned to one or more stereotypes (e.g. beginner, middle level, advanced, expert). Each stereotype has predefined properties inherited by corresponding users;
 - Overlays – the user knowledge is considered as a subset of the expert knowledge. This way it is possible to use the same presentation for the domain model and the user model;
 - Mixed – a combination of the previous two approaches.

In order to meet the eLearning requirements, these models must be explicitly present with relevant possibilities for support and control at the architectural level. An essential problem here is the explicit presentation, processing and control of their intrinsic semantics.

3.2 Approaches for Development

eLearning systems could be developed by employing:

- *Standardized architectures* – multi-usable, independent of the application area, these architectures can easily adapt to control different types of business functionality. Their main disadvantage is the low efficiency during the run-time;
- *Specialized architectures* – developed for concrete applications these architectures provide better run-time efficiency for a particular functionality but with low adaptability and portability;
- *Hybrid architectures* – a hybrid between the first two types; an adaptation (my means of reengineering, additional modules etc.) of a specialized architecture for the purposes of a particular domain.

Our approach for the development of the proposed eLearning system architecture could be categorized as a *hybrid* one. A standardized (service-oriented) architecture expanded with some adaptable components for the three models (domain-, pedagogical-, user-) is used for the eContent generation, whereas the *domain-based approach* is used as more suitable for the eContent interpretation. Special attention is paid to the architectural solutions for InfoStation-based mobile access to the system due to its inherent features such as “many-time, many-where” eLearning. Besides this the chosen approach offers enough adaptability and flexibility to the architecture, which could be controlled as:

- *Adaptable process* – with possibility for parameter control during the process of eContent generation;
- *Adaptive process* – with dynamic parameter control during the system run-time.

4. InfoStation-Based Network Architecture

The proposed InfoStation-based network architecture provides mobile access to eLearning services, for users equipped with wireless devices, via a set of InfoStations deployed in key points around a University Campus. The InfoStation paradigm can be considered as an extended wireless Internet, (Adaçal and Bener 2006), where the mobile clients interact directly with the Web service providers (i.e. InfoStations). This architecture is built upon a number of wireless communication standards [IEEE 802.11 WLAN (WiFi), IEEE 802.15 WPAN (Bluetooth), and IEEE802.16 WMAN (WiMAX)] utilized to deliver these services to the mobile devices of registered users. A three-tier network architecture is proposed with the following basic building entities (Figures 2 and 3): user mobile devices, InfoStations, and an InfoStation Center. The users request eLearning services (via their mobile devices) from the nearest InfoStation via available Bluetooth, WiFi, or WiMAX connection. The InfoStation-based system is organized in such a way that if the InfoStation cannot fully satisfy the user request, the request is forwarded to the InfoStation Center, which decides on the most appropriate, quickest and cheapest way of delivering this eLearning service to each user according to his/her current individual location and mobile device's capabilities (specified in the user profile).

Fig. 2. The 3-tier InfoStation-based network architecture (*vertical plan*)

The *First Tier* of the network architecture encompasses mobile devices (cellular phones, laptops and PDAs), equipped with intelligent agents that act as personal assistants for the users. The intelligent mobile agents onboard these mobile devices can function autonomously to accomplish tasks without human interaction.

The *Second Tier* consists of InfoStations, deployed around a University Campus, and facilitating the mobile users' access to the eLearning services through high-speed, geographically intermittent connections. In order to provide sufficient access for a great range of mobile devices, the InfoStations are equipped to support multiple communications protocols (as illustrated in Figure 3). The InfoStations can be envisaged as mobile web service agents, which act for mobile clients like gateways to services. As web service agents they receive input parameters required to execute a service, and in return could provide the

desired service. In doing this, the agents take into account the individual users' preferences and context information as specified in their profile.

The *Third Tier* is the InfoStation Center itself. Its main functions are to: control the InfoStations, update and synchronize information across the system, manage different types of profiles. The profiles contain information pertaining to a specific user or service in a way facilitating the delivery of these services in a more personalized and context-aware manner to a wide variety of devices:

- *User profiles* allow educators to monitor the progress of learners through the material, direct their learning process (ensuring the correct material will be covered), keep track of learners' test scores, and monitor their possible problem areas. The user profiles also contain information about the users location and the device they are currently using (this allows for more efficient and faster delivery of eLearning services), and information regarding the users' personal preferences (e.g. the format in which the user wishes to receive information). The preferences could also specify acceptable cost (if any) of each service, i.e. the user may wish to receive information in the cheapest way possible (Stojanov, Ganchev et al. 2003) (Ganchev, Stojanov et al. 2005) (Ganchev, Stojanov et al. 2004).
- *Service profiles* allow for easier discovery of services by the users. They also ensure that the users will not end up receiving a service not requested by them.
- *User service profiles* maintain the service status of each particular service when used by a particular user.

Fig. 3. The InfoStation-based network architecture (*horizontal plan*)

Before being distributed to the relevant InfoStations the services and their content are created/deployed in the *InfoStation Center*. It is here that the operations and maintenance of these services take place. In the case where an update of a service or a service profile is required, it is here within the InfoStation Center that this initially occurs. This update is then forwarded onto the InfoStations within the campus. The InfoStation Center also houses a repository of all profiles relating to both users and services alike. This repository maintains up-to-date copies of all profiles. Any changes made by individual users to their own user profiles are forwarded on from the user mobile device, through the InfoStation to the InfoStation Center, and the repository is updated. There are some common support functions that each service requires when initially created, for example device management,

profile management, service catalogue etc, as illustrated in Figure 4. These common functions not only act as service facilitators, but also enrich the service provision. The InfoStation Center also houses the Business Support Domain, (Andersson, Ljung et al. 2006), with a number of components relating to the billing of users, User Relationship Management (URM), and Resource Planning (RP). The URM entails all aspects of the interactions that eLearning system has with its users, whereas the RP is a management system, which integrates all facets of the business side of the service provision.

The *InfoStations* are deployed throughout the University Campus area and provide wireless access to eLearning services. When a user mobile device comes within the range of an InfoStation, both mutually discover each other. Then the InfoStation authenticates, authorizes and accounts (AAA) the user. To do this the InfoStation must maintain up-to-date copies of all user profiles, as well as a catalogue of all services. Once the users profile is analyzed, the InfoStations discerns which services from the service catalogue are applicable to that specific user, taking into account the user preference and mobile device capabilities. The InfoStation then forwards an updated list of the relevant services to the intelligent agent working as a personal assistant on the user mobile device. When this assistant forwards the user request for a particular service to the InfoStation, the InfoStation instantiates the service, allowing the user an access to it. If the InfoStation is unable to facilitate the user service request, it forwards this request to the InfoStation Center to deal with this. Figure 4 visualizes in more detail the 3-tier architecture of the system by utilizing some of the components proposed in (Andersson, Freeman et al. 2006).

Fig. 4. The InfoStation-based network architecture

Within the mobile user domain (i.e. the user mobile device), the personal assistant (agent) allows the user to utilize the service, by migrating the service to the users device. This is down to the agent-oriented approach to the implementation of the system, which is discussed in the next section. With the service migrated onto the users mobile device, the user may utilize the service even when out of range of the InfoStation. The personal assistant may make a service request while within the range of an InfoStation, and then pass

out of its coverage area. The personal assistant however will continue to work autonomously, adopting the functionality of the service until the user has completed his/her task (e.g. an eTest). Once the mobile device comes within range of another InfoStation, the service instance is updated and synchronized to reflect any eLearning work completed by the user while out of range.

5. Multi-Agent System Structure

In the previous section we described the generic (conceptual) InfoStation-based network architecture and presented its common functionality for the provision of eLearning services within a University Campus. In this section we discuss the software architecture required to provide the needed hardware flexibility and adaptability according to the requirements of modern eLearning systems. The software architecture could be considered as a singular mapping of the *conceptual model* (section 3.1) over the InfoStation network architecture. Subsequently the software could be written as a system of cooperating *agents* and electronic services (*eServices*) – a solution justified in this section. As a starting point for a step-by-step definition and specification of the software architecture we first consider the possible scenarios in the InfoStation infrastructure.

5.1 Scenarios

Due to the inherent mobility afforded by the InfoStation, it is quite possible that not only will the users move from one geographically intermittent InfoStation cell to another (*device mobility*), but also from one mobile device to another (*user mobility*). The main problem here is the distribution of the session control across the three tiers of the InfoStation architecture (by taking into account the possibility of changing mobile devices and InfoStations during the session execution). The following main scenarios are possible:

- *No change* – the simplest scenario in which the user mobile device and the serving InfoStation are not changed during the eLearning service session (i.e. the user is relatively stationary);
- *Change of InfoStation* – the serving InfoStation is changed during the eLearning service session (i.e. the user is on the move), but the user mobile device remains the same;
- *Change of user mobile device* – the user mobile device is changed during the eLearning service session, but the serving InfoStation remains the same (e.g. the user service session is switched from a mobile phone to a laptop; the user may move around but remains within the range of the same InfoStation);
- *Change of InfoStation and user mobile device* – the most complicated scenario where both the user mobile device and the serving InfoStation are changed during the user service session.

In general the user service session may start with one set of devices, may continue with another, and may finish with a third set of devices, for example a user driving to work may begin a session using an onboard computer or mobile device, then whilst on foot s/he may continue the session on her/his mobile device, and finally once the user arrives at the office, s/he may choose to complete the session using a desktop PC. The possibility of such scenario sets a requirement for an effective distribution of session control across the agents.

5.2 Control & Management

Proper solutions for service control and user management are of high importance for any eLearning system architecture. As stated earlier, specialized components for explicit control within and between the three models (domain-, user-, pedagogical-) need to be envisaged. The *user model* and the *pedagogical model* are directly related to user management and could be used also for parameterization of eLearning services. The *domain model* is used primarily by the eContent services and could be used also for user management in the *hybrid architecture* approach. These problems are treated in detail in this section.

5.3 Service control

Means for proper service control are needed in every software architecture. In our proposed architecture the concept of service control is developed around a service classification model presented in Figure 5. This model serves the development of the system managing the service invocation, which is part of the system run-time module. The service model is open for extensions including new values of already specified dimensions or entirely new dimensions. In order to separate the architecture from the offered services, the service classification model is developed as independent from the infrastructure. The model consists of two main components: service characteristic space (*meta-model*) and *subject models*. The meta-model classifies services within an n -dimensional discrete hyperspace.

Fig. 5. The service classification model

For the moment four characteristic dimensions are defined:

- *Content/nature* – a classifier of services according to their content/nature, i.e. eLearning, eEducator, Administration, and System services;
- *Invocation* – a classifier of services according to the activation option provided in the infrastructure, i.e. Local, Remote, and Back-End services;
- *Mobility* – specifies whether the service is a mobile one or a stationary one;
- *Standard-conformance* – whether the service is conformant (or not) to an eLearning standard, e.g. SCORM(SCORM 2006), ARIADNE(ARIADNE), AICC (AICC), IEEE LTSC (LTSC), IMS (IMS), CEN/ISSS (CEN/ISSS) etc.

The subject models specify the functionality of the offered eLearning services. Besides this classification for control purposes each service is considered as consisting of two parts:

- Service's user part (SUP) - an interface used for service control during initialization (searching, discovering, identification...) and finalization (in case of device change). SUP is deployed in the mobile devices and the InfoStations;
- Service's server part (SSP) - deployed and executed by the only non-changeable component - the InfoStation Center. It is in the SSP that the functionality provided by a service is realized..

This logical (conceptual) service structure could be mapped to different models and protocols for service control. Some services (mainly eLearning and eEducator services) directly use knowledge presented in the *domain model*. From an architectural point of view the possibility for dynamic switch/change (during the run-time) from one learning area to another is very important. Today's approaches for the implementation/realization of eLearning models usually use *ontologies*. Besides this, depending on their nature, some services could be parameterized by retrieving the parameter values from the *user model* and the *pedagogical model*. This increases the possibilities for architectural support for one of the main characteristics of the eLearning systems, namely service personalization. The possibility for parameterization, parameter types, their allowable values, the time needed to assign these values (during service development or dynamically during the run-time) are defined in the corresponding subject models.

5.4 User management

As a basis for the development of the concept of user management we use the *stereotype approach*, presented in section 3.

Fig. 6. Agent-oriented approach for user support

An essential problem, which needs to be taken into account when implementing the architecture, is that the users of particular services would be known only at the run-time. Hence for real personalization, the architecture must have enough flexibility in order to support both aspects of adaptability – being adaptable and being adaptive:

- *Adaptable* – server reference agents, corresponding to a particular user model, are configured and deployed a priori in the server part of the architecture (InfoStations, InfoStation Center). These agents present given stereotypes to the potential users;
- *Adaptively* – an appropriate stereotype agent will be chosen for each new user during his/her first access to the system. From the chosen stereotype and the user personal profile a corresponding personal assistant will be generated as an agent-instance of the server agent. Then the personal assistant will be transferred in the user-operating environment. Subsequent efficient and adaptive user accesses to the system are possible through the use of the personal assistant, Figure 6.

The proposed approach for user management could be used also for the implementation of the *mixed user models*. In this case ‘more intelligent’ agents, aware of the users’ background knowledge, are needed (so as to reflect the actual knowledge level of learners in particular area of study). For the management of mixed models corresponding *domain models* could be also used.

Another important problem is related to user session management. Known client-server management mechanisms cannot be used in our architecture due to difficulties in tracing the user sessions in mobile scenarios discussed previously. For user session management we propose to use a goal-driven approach, whereby in correspondence with the used pedagogical model the actual learning tasks/goals are defined during the learning process itself. These goals are recorded in the actual user profiles and their fulfilment could be traced by the server part of the architecture and/or by the learners’ personal assistants. With the help of their assistants these users could interrupt/suspend their service sessions (e.g. without achieving particular learning goal, if allowed).

5.5 Agent control models

An analysis of the expected behaviour of all components is needed when designing eLearning system architectures. The architectural components could be categorized as follows:

- *Passive* – only offering particular functionality without being able to activate services and control their execution;
- *Reactive* – can activate and control a particular functionality/service as a response to an external request;
- *Proactive* – can make self-dependent autonomic decisions for activation and execution of a particular functionality/service.

Careful analysis of the previously considered scenarios and the resulting distribution of functions and responsibilities between the different tiers of the proposed architecture suggests that it is reasonable to implement the software for passive and reactive components as electronic services (*eServices*), whereas for the software implementation of the proactive components *agents* are required.

For implementation of the proposed InfoStation-based eLearning system architecture as a combination of agents and eServices we have developed two possible models: ‘*thin agent*’ model (Figure 7) and ‘*thick agent*’ model (Figure 8). For effective software support of the

As a result the interoperation between an InfoStation and the InfoStation Center is 'agent-to-eService' (e.g. based on OWL-S protocol), i.e. the InfoStation agents see the InfoStation Center as a set of different eServices. The main disadvantage of this model is the substantial load on InfoStations with regard to control tasks. The reduced coordination between the InfoStation and InfoStation Center is certain advantage of this model. Independently of the model chosen for the actual system implementation (*thin-agent* or *thick-agent*) it is necessary to solve the problem of communication between the agents themselves on one hand, and between the agents and eServices on the other. This type of communication could be realized:

- By using the existing communications standards/protocols, or
- By choosing (and mixing) suitable subsets of given communications standard/protocol;
- By developing a new communication protocol.

Each of these three approaches has its own pros and cons thus the choice must be carefully thought and based on detailed comparing analysis. For communication between agents currently we are analyzing the communication capabilities of the Agent Communication Language (ACL) (FIPA 2002), developed as a part of the Foundation for Intelligent Physical Agents' (FIPA) specification, or its limited subset. The ACL defines the messages used in the communication between intelligent agents within this system. Abstract messages passed between intelligent agents within the system consist of 2 components: an envelope (containing information utilized in the transportation of the message) and the message payload. The ACL is not concerned with the envelope or specifying how a message is disseminated throughout the system, but rather it defines the payload or representation of the information contained within the message itself. The ACL defines a number of parameters which are utilized to provide effective agent communication. Depending on the type of communication, which parameters are utilized will vary. Only one of the parameters is mandatory in all ACL messages, this being the *performative*. This parameter defines the type of communication that particular message performs. The performative signifies if a message is a *request*, a *proposal*, or indeed a *query*.

As well as a *performative* parameter, most messages passed between the agents within the system will also contain parameters which are utilized to specify the various participants in a communication. The *sender* parameter, for example, will be contained within most messages passed between agents in this system. This parameter refers to the agent who initiates the communicative act. The *receiver* parameter denotes one (unicast) or more (multicast) destinations for a message. The *reply-to* parameter could prove most useful within this system. During a 'conversation' between agents, by specifying an agent within the *reply-to* parameter, subsequent message are sent on to this agent instead of the agent specified with the *sender* parameter. So far we have detailed parameters which outline the type of ACL message and the parties involved within a particular communication. However, ACL parameters can also be utilized to specify the content of the message, as well as a description of the content. The *content* parameter denotes the actual content of the ACL message itself. Within these ACL messages, this content could refer to any of the information passed back and forth between the PA and the InfoStation and depending on the agent model chosen, the intelligent agent operating within the InfoStation Center. For example, profile update information, user service requests, service content information etc. For communication with eServices the agents need computer-understandable description of the eServices, and instructions of how to interoperate with the eServices. A good option here

is offered by the Ontology Web Language (OWL-S) protocol (Ankolekar, Burstein et al.). To use an eService, the OWL-S provides a set of constructs used to create *ontologies* (i.e. specifications of concepts and relationships with the agent), which are machine-understandable descriptions of the service. The ontology structure is divided up into three separate sections, each dealing with a different aspect of the agent:

- *Service Profile* - advertises the abilities of the service (i.e. what it can do) by providing a concise high-level description of a service to a registry. Up-to-date copies of service profiles are kept in the InfoStation Center. The InfoStation Center periodically delivers copies of the service profiles to all InfoStations. The service profile is used for populating service registries, automated service discovery, service selection etc;
- *Process Model* - gives a detailed description of the service operation and tells a user how and when to interact with a service (read/write messages). This model is used for service invocation, planning, composition, interoperation, monitoring etc;
- *Grounding* - provides details of how the agents can interoperate/interact with a service, e.g. describes message formatting, transport mechanisms, protocol types, serializations etc. When combined with the process model, the grounding gives everything required to utilize a service.

Agents utilize the information contained within the Service Profile to ascertain whether or not a service meets its requirements, and adheres to certain constraints such as security, and quality etc. While the Service Profile provides all the information needed for an agent to discover a service, the Process Model provides the information necessary for the agent to use the service. The Process Model allows the agent to perform a more in-depth analysis of the service and its capabilities, and determine if it can be utilized. As well as this, it enables the agent to compose new service descriptions through the composition and interoperation of previous existing services, to perform specific tasks. The Service/Process Model also allows agents to monitor the execution of tasks performed by a service (or a set of services), and to coordinate the entities involved in the service execution. The Service Grounding details how agents can communicate with, interoperate with, and invoke a service. The relationships between the various service components are modelled using properties such as *presents* (Service to Profile), *describedBy* (Service to Process Model) and *supports* (Service to Grounding), as illustrated within Figure 9.

Fig. 9. OWL-S component relationships (Martin, Burstein et al. 2004)

These three parts altogether create an ontology/description, which allows intelligent agents to discover, invoke, compose and monitor eServices. The OWL-S protocol offers a

good opportunity for the realization of flexible software architectures and offers a suitable environment for the support of the intelligent mobile services envisaged in our architecture. We treat the OWL-S specification in a distributed fashion, where the exact scheme of distribution depends on the chosen model – *thin-agent* or *thick-agent*.

5.6 Agent types

The agents we use in our eLearning system architecture are divided in two groups:

- *Personal Assistants* – mainly used to support users when using the system. These agents are visible to the users. Particular attention is paid to the development of intelligent, effective and user-friendly GUI;
- *Server Agents* – mainly used to support the operation of the InfoStations and the InfoStation Center, by assuring proper communication between personal assistants themselves on one hand and activation of the desired eServices on the other. These agents are transparent to the users. Main attention paid in their implementation is related to their effectiveness during operation. They do not have user interfaces of any kind.

Besides this classification the agents may have different level of intelligence:

- *Goal-oriented agents* – used to support the *pedagogical model* and the *mixed user model*;
- *Mobile agents* – for support of the proposed implementation scheme of the *user model*.

6. mLearning Services

Some examples of services used in our system are considered in the following subsections, illustrating the sequences of events which take place during service execution.

6.1 mLecture

The mLecture service exists as the very core service within this system. While initially this service would be used as a supplementary aid to the traditional learning experience. In a traditional learning environment, it is often the case that a lecturer will make available various ancillary learning resources in order to aid the students understand the presented information. Usually this would consist of a set of notes to accompany the lecture. In this case, a lecturer could upload these learning resources (e.g. text, diagrams, images) pertaining to the lecture, or indeed supply a pod-cast or possibly even a video-cast of the lecture itself. This would greatly assist the student's assimilation of information. As students would be able to regulate the pace at which they proceed through the information and if necessary re-cycle back through the lecture. This ensures the material be more accessible for learners of varying learning styles. For some time now pod-casts have been utilized around the world in third level institutions to aid students. Building on these examples the mLecture service seeks to adopt a more multimedia approach to course delivery (where possible), incorporating audio, video and elements of interactivity to the process. Indeed while this service would greatly aid the success of the traditional learning paradigm, the ultimate goal of employing a service such as this would be the eventual presentation of a standalone distance learning course. Indeed this seems more then feasible when we consider the delivery of video-cast lectures. In the mean-time this technology serves to greatly enhance the traditional teaching/learning environment.

Fig. 10. *mLecture* service provision

The sequence diagram Figure 10, visualises these various entity interactions, including the situation which arises if the local InfoStation does not have a copy of the most appropriate service format within its service repository. The initial interactions, when the mobile user first enters within the range of an InfoStation, are concerned with the processing of the user's AAA request, and subsequent selection of service. In this case the service chosen is the *mLecture* service. The PA forwards the user's service request to the InfoStation. If the

InfoStation does not have an up-to-date cached copy of the required mLecture, it forwards the request onto the InfoStation Center, which in turn instantiates the service and sends back to the InfoStation a full copy of the requested lecture. The InfoStation examines the user and user service profiles and customizes the service accordingly. The user's current device may not possess the capabilities to accessing the full range of media formats available as part of a service. For this reason, the InfoStation will adapt the service content to a format which 'best' suits that particular device (i.e. accounting for both the device and the access network) according to the specifications within the User Profile.

6.2 mTest

A key component for a successful eLearning process, mTest provides a means to evaluate the students acquired knowledge and provides valuable feedback to students concerning their progress. mTest also allows the educator to shape the learning experience of the students, ensuring the student remains engaged in the correct material. Indeed the main benefit of using quizzes is the motivation of the students engagement in the material, without the stress of associated with traditional exams (CAA). The utilization of short quizzes throughout the duration of a course can also have the added benefit of providing great motivation to students. By providing feedback on their progression, student can be made aware of how well they are assimilating the presented course content. Educators may also benefit from such information. By monitoring the progression of a group of students, the educator may actively modify their approach to conveying the course content and as such, optimize the performance of the group, and enhance the overall learning experience. Indeed this process can be even further enhanced through the use of pervasive technologies such as Short Messaging Service (SMS). SMS quizzes have proved over the last few years, to be a very useful tool in education (especially within third level institutions). For example (Markett, Arnedillo Sánchez et al. 2004) discusses the uses of SMS to support interactivity within the classroom. With this, the students used their personal mobile phones during class to send SMS in real-time to a modem accessible by the lecturer in real time. So during the course of the lecture, the lecturer can dynamically address the SMS content as it arrives. This allowed the students to become more actively involved in the class, without disrupting it. The sequence diagram, Figure 11, depicts sample interaction between entities involved in this service. As outlined previously, when a mobile user enters within the range of an InfoStation, s/he goes through the normal AAA procedure, and selects the relevant service. In this case the *mTest* service is chosen. Once the user/student gains access to the service, s/he may choose a particular assessment. The PA on the user's mobile device forwards a service request on to the InfoStation, specifying the user's choices. The InfoStation in turn, having analysed the capabilities of the target device, discerns the optimal format in which to present the assessment. This service content is then made available to the PA. As the student progresses through the test, his/her user service profile is maintained to reflect this progress. Furthermore we consider the possibility for the student to do the test whilst on the move and out of range of an InfoStation. Due to the geographically intermittent nature of an InfoStation connection, the student's mobile device may leave the contact range of the initial InfoStation (InfoStation 1). However the PA facilitates the users continued utilization of the service while at the same time maintaining the user service profile. Thus the student may complete the test whilst outside the signal range of any InfoStation, with the user service profile reflecting the student's progression through the material. InfoStation 1, the last

InfoStation in contact with the PA, sends user profile and user service profile updates to the InfoStation Center, ensuring it has the most up to date information.

Fig. 11. *mTest* service provision – Change of InfoStation

Once the PA eventually does enter back within range of an InfoStation (i.e. InfoStation 2), the PA will again go through the process of authentication, authorization and accounting. The PA forwards on a user service profile update which reflects the progress of the student through the service content, whilst out of range of the InfoStation network. These updates

are disseminated through to the InfoStation Center so as to ensure all information across the system is up-to-date. To the user, this transition between InfoStations should appear seamless, with the user not experiencing any loss of service. Once the user has completed the mTest, the PA displays the results of the assessment to the user, providing valuable feedback on their own progress and performance. A notification of service termination is forwarded on by the PA to the local InfoStation, along with the user profile and user service profile updates. Contained within these updates is information relating to the various grades achieved by the user in any assessments undertaken by the user in that particular service session. This information is in turn accessible by the lecturer/educator, allowing them to monitor the performance of the students.

7. Implementation

7.1 Composite Capability/Preference Profile (CC/PP) & User Profiles

One of the main issues concerning the implementation of the eLearning services is related to the creation of user- and service profiles. As we have mentioned previously, the Ontology Web Language (OWL-S) is utilized in the creation of the service profiles, which facilitate the advertisement of the services and interoperation between the agents and these services. We propose to use the "Composite Capabilities/ Preference Profile" (CC/PP) (Tran, Butler et al. 2003) as an uniform format for the implementation of the user and mobile device profiles. Recommended by the World Wide Web Consortium (W3C) (W3C) this platform-independent format is based on the Resource Description Framework (RDF) (Manola and Miller 2004). The CC/PP basically provides a standard way for to make known devices capabilities and user preferences. When a mobile device sends a service request to an InfoStation / InfoStation Center using the CC/PP, the latter can tailor the content of the service to suit the requesting device. In essence the content is adapted to suit the individual user. Therefore, eLearning services can be provided to users, regardless of mobile devices currently in use. As such, the eLearning service developers can create device-independent code and not worry about the mechanisms users will utilize to gain access to these services. The majority of CC/PP-enabled devices utilize the User-Agent Profile (UAProf) (UAProf-working-group 1999). This concrete implementation of the CC/PP, developed by the Open Mobile Alliance (OMA), enables the flow of capability and preference information between an Internet-enabled mobile devices and a requesting server. The UAProf is aimed at WAP-enabled mobile devices, and defines a profile vocabulary that describes the hardware and software characteristics as well as the WAP-specific features of the mobile device and associated transport network capabilities. CC/PP and UAProf represent the devices capabilities in the form of a two-level hierarchy consisting of components and attributes. Components are a grouping mechanism for attributes, therefore in essence, CC/PP or UAProf profiles are organised as structured sets of attributes and value pairs. When a user's mobile device enter the range of an InfoStation, they will mutually discover each other and the intelligent agent on the users device will forward a UAProf to the InfoStation. From this profile the InfoStation can dynamically adapt and customize services to suit that device. However to customize the service to suit the preferences of the user, the profile must be extended to specify these preferences. This is achieved through the extensibility of the CC/PP vocabulary, which defines the format/structure of the profile information exchanged between a client and a server. This information can be extended to describe some

important attributes of the user too. The user preference components can specify anything from the user's name and the languages s/he speaks, to her/his location and the preferred format in which the user wishes to receive information. An important attribute within the user profile, especially within the educational domain, would be to specify the role of the user, whether the individual is an educator or a learner. While these pre-defined UAProf attributes allow a service to be adapted and tailored to suit the operating environment on a particular device, it's also necessary to express our own additional components and attributes in order to allow the right services to be offered to the right users. Fortunately the use of RDF allows for the extension of UAProf schemas to suit the requirements of this system. As such, Profiles will be composed not only of components and attributes from a base vocabulary, but also from our own specific schema extensions. Within these extended schemas, we can specify a multitude of attributes relating to a wide variety of user's. As we've mentioned, the role of a specific user is very important, depending on whether the user is an educator, a student, or indeed a visitor to the campus will have a bearing on the levels of access the user has to services and service content. This information allows the services to provide a more personalized and higher quality of service to users. Figure 8 is an example of how a component and group of attributes relating to a particular user may specify vital information about that individual.

```
<prf:component>
<rdf:Description rdf:ID="UserPlatform">
  <rdf:type rdf:resource="http://www.ece.ulie/trc/profiles/UAPROF/ccppschem-1#UserPlatform" />
<prf:Name>John Doe</prf:Name>
<prf:StudentID>0123456</prf:StudentID>
<prf:Faculty>ECE</prf:Faculty>
<prf:Course>Electronic Engineering</prf:Course>
<prf:Year>4</prf:Year>
<prf:Classes>
  <rdf:Bag>
    <rdf:li>CE4517</rdf:li>
    <rdf:li>CE4607</rdf:li>
    <rdf:li>CE4717</rdf:li>
    <rdf:li>CE4817</rdf:li>
    <rdf:li>CE4907</rdf:li>
    <rdf:li>EE4607</rdf:li>
  </rdf:Bag>
</prf:Classes>
<prf:Advisor>Dr. Ivan Ganchev</prf:Advisor>
<prf:email>0123456@STUDENT.ulie</prf:email>
<prf:QCA>3.47</prf:QCA>
<prf:FYP>JD09</prf:FYP>
<prf:FYPSupervisor>Dr. Ivan Ganchev</prf:FYPSupervisor>
<prf:FYPTitle>Design and Implementation of an Animated Interactive Tutorial</prf:FYPTitle>
</rdf:Description>
</prf:component>
```

Fig. 12. Example of a CC/PP User specific Component

Within this sample profile component, we can see some attributes which will prove essential in order to properly advertise services to the most applicable users, and indeed adapt them to suit this particular user. In the given sample component, Figure 12, an attribute such as 'Faculty' can have a major bearing on the type of services being offered to a particular user. For example the services utilized by members of the Business faculty will generally differ

greatly from those being utilized by members of the Electronic and Computer Engineering faculty. Indeed by allowing for the specification of the various classes being taken by the user, services can be directed at even more specific ranges of users. It's essential for these factors to be taken into account in order to avoid unnecessarily advertising irrelevant services to users. In the case of charging and billing, the user may request access to services in the cheapest manner possible or indeed with the highest possible quality. As the preferences of the individual will be as varied as the individuals themselves, a mechanism within the Intelligent Agents (Personal Assistants), on-board the mobile devices, will allow the individual users to alter their own preferences.

7.2 J2ME & Device Independence

In the modern mobile environment, there are many diverse device types characterized by different screen sizes, input (keyboard, touch, keypad etc.) and other characteristics. A standard, platform-independent architecture is required for the creation of mobile applications across these diverse devices. The standard architecture we use for development, deployment and execution of the intelligent mobile services is the Java 2 Micro Edition (J2ME) (JSR-Groups). J2ME provides a modular, scalable architecture that allows for the flexible deployment of Java Technology on a multitude of different devices with different features and functions. J2ME is basically a collection of building blocks and frameworks that can be combined to suite particular devices sharing similar characteristics. J2ME's device independence stems from its use of profiles, configurations and other optional packages.

A J2ME "configuration" targets devices with a specific range of capabilities and defines the minimum elements required by those devices (i.e. a given JVM specification, a core API). A J2ME "profile" selects a configuration, and then defines a specific set of API's aimed towards a specific set of services/abilities that support a certain category of devices, within the framework of that configuration. By selecting the appropriate configurations and profiles, applications can be developed and deployed regardless of the device onto which they're to be deployed. The Connected Limited Device Configuration (CLDC) (JSR-139 2003) is a fundamental part of the J2ME architecture. It provides the most basic core set of libraries and virtual machine features that must be present within each implementation of a J2ME environment, i.e. it provides a reduced JVM implementation. This ensures low memory and CPU consumption, which are especially important factors when developing applications for mobile devices with limited resources.

The Mobile Information Device Profile (MIDP) (JSR-118 2002) defines a platform for dynamically and securely deploying of applications and services on mobile devices. It provides support for a graphical interface, networking and storage of persistent data for mobile device applications called MIDlets (similar to applets). When these profiles and configurations are combined they provide a complete Java application environment for a specific device class. J2ME provides a number of optional packages, one of the most useful of which is the Wireless Messaging API (WMA) (JSR-120 2003). This API provides platform-independent access to wireless communication resources such as the Short Messaging Service (SMS). The WMA is used on top of CLDC and MIDP that provide the core functionality required by mobile applications.

7.3 J2ME & Device Independence

Our proposed agent-oriented system is implemented by using the Java Agent DEvelopment (JADE) framework developed by TILAB (Bellifemine, Caire et al. 2005) (Bellifemine, Caire et al. 2003). This allows flexible development of multi-agent systems and applications for management of networked resources in compliance with the FIPA specifications. "JADE is an enabling technology, or middleware for the development and run-time execution of peer-to-peer applications which are based on the agents paradigm and which can seamlessly work and interoperate both in wired and wireless environments" (Bellifemine, Caire et al. 2003). JADE has been fully implemented in Java, as Java provides many features geared towards object-oriented programming in distributed environments, such as Remote Method Invocation (RMI) and object serialization. JADE provides a set of APIs completely independent of the underlying network and the Java version (the same APIs are provided for each different edition of JAVA - J2EE, J2SE, J2ME). This illustrates JADE extreme versatility for being able to be integrated into complex structures such as J2EE; it can also be tailored to fit the constraints of environments with much greater limitations on resources. For example, the JADE run-time memory footprint in a MIDP 1.0 environment can be as between 50 - 100KB. This limited memory footprint allows JADE to be installed on any Java-enabled mobile devices. This versatility also allows developers to re-use the same application code, whether it is deployed on a PC, a PDA, or a Java-enabled phone.

Each instance of the JADE run-time is called a *container*, which may contain a number of agents. A group of active containers together is called a *platform*. A single *main container* must always be active on every platform. The first container to become active on a platform assumes this role, and all other "normal" containers must register to this main container as soon as they become active on the platform. The main container also differs in that it contains two special agents, which automatically start when it is launched: the Agent Management System and the Directory Facilitator agents.

As with Java architecture, the JADE architecture is completely modular. By utilizing specific modules, JADE can be configured to adapt to the requirements of the deployment environment (e.g. processing power, memory, and connectivity). For our implementation of JADE, one of the most important modules is the Light Extensible Agent Platform (LEAP) module (Caire and Pieri 2006). This module replaces some parts of the JADE kernel providing a modified run-time environment for enabling FIPA agents to execute on a wide range of Java-enabled devices. It allows for the optimization of communication mechanisms when dealing with mobile devices (with limited resources) connected through wireless networks. A container is split into a *FrontEnd* (running on the mobile device itself), and the *BackEnd* (running from a fixed network entity called a *mediator*), as illustrated in Figure 13.

Fig. 13. FrontEnd and BackEnd of a "JADE powered by LEAP" implementation

This mediator is charged with instantiating and maintaining the BackEnds. In our system, the InfoStations deployed throughout a University Campus take on this mediator role. Each FrontEnd is connected to each BackEnd through a wireless bi-directional connection. The splitting of the container into two separate, yet connected entities is particularly useful in the realm of resource constrained devices, as the FrontEnd of the container is of far more lightweight in terms of the required memory and processing power than the entire container. Due to the geographically intermittent nature of the InfoStation connection, the FrontEnd and the BackEnd may undergo a loss of connection; however the front-end can detect this and re-establish the connection as soon as possible. Any messages not transmitted due to this temporary disconnection can be buffered and delivered when the connection is re-established. This store and forward mechanism is implemented in both the front-end and the back-end, and is vitally important for the InfoStation-based eLearning system architecture.

This splitting of the container has no bearing on the application developers, as the same functionality and set of APIs is available to an agent, whether it is contained within a full container or the front-end of a split container. The JADE framework also serves to shield the developers from the complexity of the distributed environment, allowing them to concentrate their efforts on developing the application logic, rather than worrying about middleware issues such as discovery and communication of entities within the system.

Within JADE, communication between the agents is facilitated through the use of the Agent Communication Language (ACL).

8. Conclusion

The realization of InfoStation-based system architecture for the provision of intelligent mobile services in a University Campus area has been outlined within this chapter. A detailed description of the 3-tier InfoStation-based network architecture along with its functional entities has been presented. A step-by-step multi-agent approach for the definition and the specification of the software architecture has been used by first considering the main scenarios in the InfoStation infrastructure, followed by the proposed solutions for service control and user management, and finishing with a justification for the software implementation of the passive and the reactive components of the architecture as electronic services (*eServices*), and as *agents* for the proactive components. Two examples of mLearning services supported by this underlying architecture were presented; highlighting the interactions between the main system entities during a service session. The Composite Capabilities/Preference Profile (CC/PP) standard has been adopted as a uniform format for the implementation of the user/service profiles. The Java 2 Micro Edition (J2ME) has been used as the standard architecture for the development, deployment and execution of the proposed intelligent mobile eLearning services. The proposed system is organised according to a multi-agent structure, and is implemented by means of the Java Agent Development (JADE) framework, making particular use of its LEAP module.

9. References

- Adaçal, M. and A. Bener (2006). "Mobile Web Services: A New Agent-Based Framework." IEEE Internet Computing Vol. 10 (no. 3): pp. 58-65.

- AICC Aviation Industry CBT Committee at <http://www.aicc.org/>.
- Andersson, C., D. Freeman, et al. (2006). *Mobile Media and Applications - From Concept to Cash: Successful Service Creation and Launch*, John Wiley & Sons.
- Ankolekar, A., M. Burstein, et al. OWL-S: Semantic Markup for Web Services, DAML Program.
- ARIADNE Alliance of Remote Institutional Authoring and Distribution Network for Europe (ARIADNE) at <http://www.ariadne-eu.org/index.html>.
- Bellifemine, F., G. Caire, et al. (2003). "JADE: A White Paper." exp, Telecom Italia Lab Volume 3 (No. 3).
- Bellifemine, F., G. Caire, et al. (2005). *JADE Programmers Guide*, TILab.
- Biggs, J. B. (1999). *Teaching for Quality Learning at University: What the Student Does*. Philadelphia, Society for Research into Higher Education.
- CAA. "Computer-Assisted Assessment Centre at <http://www.caacentre.ac.uk/index.shtml>." from <http://www.caacentre.ac.uk/index.shtml>.
- Caire, G. and F. Pieri (2006). *LEAP User Guide*, TILab.
- CEN/ISSS. "CEN/ISSS Learning Technologies at <http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/index.asp>." from <http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/index.asp>.
- Cook, S., R. Harrison, et al. (2003). *Challenges of Highly Adaptable Information Systems*, Applied Software Engineering Research Group, University of Reading.
- FIPA. "Foundation for Intelligent Physical Agents (FIPA) - <http://www.fipa.org>." from <http://www.fipa.org>.
- FIPA (2002). *FIPA ACL Message Structure Specification*. Geneva, Switzerland, Foundation for Intelligent Physical Agents.
- Frenkiel, R. and T. Imielinski (1996). *Infostations: The joy of 'many-time, many-where' communications*, WINLAB Technical Report.
- Ganchev, I., S. Stojanov, et al. (2005). *Mobile Distributed eLearning Center*. Fifth IEEE International Conference on Advanced Learning Technologies, ICALT 2005, Kaohsiung, Taiwan.
- Ganchev, I., S. Stojanov, et al. (2006). *An InfoStation-Based University Campus System for the Provision of mLearning Services*. IEEE-ICALT 2006, Kerkrade, The Netherlands.
- Ganchev, I., S. Stojanov, et al. (2004). *Enhancement of DeLC for the Provision of Intelligent Mobile Services*. 2nd International IEEE Conference on Intelligent Systems (IS'2004), Varna, Bulgaria.
- IMS IMS Global Learning Consortium, Inc. (IMS) at <http://www.imsproject.org/>.
- JSR-118 (2002). *Mobile Information Device Profile (MIDP) Specification*, Sun Microsystems's.
- JSR-120 (2003). *Wireless Messaging API (WMA)*, Sun Microsystems's.
- JSR-139 (2003). *Connected Limited Device Configuration (CLDC) Specification*, Sun Microsystems Inc.
- JSR-Groups. "Java 2 Platform, Micro Edition (J2ME) - <http://java.sun.com/j2me/index.jsp>." from <http://java.sun.com/j2me/index.jsp>.
- LTSC. "IEEE Learning Technology Standards Committee (LTSC) at <http://ieeeltsc.org/>." from <http://ieeeltsc.org/>.
- Manola, F. and E. Miller (2004). *RDF Primer (W3C Recommendation)*. B. McBride, W3C.

- Markett, C., I. Arnedillo Sánchez, et al. (2004). PLS Turn UR Mobile On: Short Message Service (SMS) Supporting Interactivity in the Classroom. IADIS Conference on Cognition and Exploratory Learning in Digital Age, Lisbon.
- Martin, D., M. Burstein, et al. (2004). OWL-S: Semantic Markup for Web Services. M. D, W3C.
- SCORM (2006). Sharable Content Object Reference Model (SCORM) 2004 3rd Edition Overview, Advanced Distributed Learning (ADL).
- Stojanov, S., I. Ganchev, et al. (2003). DeLC - Distributed eLearning Center. 1st Balkan Conference on Informatics BCI'2003, Thessaloniki, Greece.
- Tran, L., M. Butler, et al. (2003). Composite Capability/Preference Profiles (CC/PP) Processing Specification. L. Tran and M. Butler, Sun Microsystems, Inc.
- UAProf-working-group (1999). Wireless Application Group - User Agent Profile Specification, Wireless Application Protocol Forum.
- W3C. "World Wide Web Consortium (W3C) - <http://www.w3.org/>." from <http://www.w3.org/>.

Advanced Learning

Edited by Raquel Hijn-Neira

ISBN 978-953-307-010-0

Hard cover, 444 pages

Publisher InTech

Published online 01, October, 2009

Published in print edition October, 2009

The education industry has obviously been influenced by the Internet revolution. Teaching and learning methods have changed significantly since the coming of the Web and it is very likely they will keep evolving many years to come thanks to it. A good example of this changing reality is the spectacular development of e-Learning. In a more particular way, the Web 2.0 has offered to the teaching industry a set of tools and practices that are modifying the learning systems and knowledge transmission methods. Teachers and students can use these tools in a variety of ways aimed to the general purpose of promoting collaborative work. The editor would like to thank the authors, who have committed so much effort to the publication of this work. She is sure that this volume will certainly be of great help for students, teachers and researchers. This was, at least, the main aim of the authors.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ivan Ganchev, Stanimir Stojanov, Mairtin Odroma and Damien Meere (2009). Development of InfoStation-based and Contextaware mLearning System Architectures, *Advanced Learning*, Raquel Hijn-Neira (Ed.), ISBN: 978-953-307-010-0, InTech, Available from: <http://www.intechopen.com/books/advanced-learning/development-of-infostation-based-and-contextaware-mlearning-system-architectures>

INTech

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.