

© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

A Middleware for Mobile and Ubiquitous Learning Ecosystems based on a Reconfigurable Plug-and-Play Architecture: Application to Mashups

Sergio Martin, Manuel Castro

Electrical and Computer Department

UNED – Spanish University for Distance Education

Madrid, Spain

smartin@ieec.uned.es, mcastro@ieec.uned.es

Alex Talevski, Vidyasagar Potdar

Digital Ecosystems and Business Intelligence Institute

Curtin University of Technology

Perth, Australia

A.Talevski@curtin.edu.au, V.Potdar@curtin.edu.au

Abstract—This paper describes a middleware intended to support the development of mashup applications in mobile and ubiquitous learning environments. It is based on a framework that integrates different contextual information (e.g. geographic location, profile, history) and services (from e-learning platforms or other learning tools). The integration of the services into the middleware is carried out through a reconfigurable plug and play architecture based on Web Services. It provides a better and more flexible management of the involved services in the system.

Keywords-middleware; service-oriented; reconfigurable plulg-and-play architecture; mashup; mobile learning; ubiquitous learning

I. INTRODUCTION

There is no doubt that one of the main issues involved in the development of mashup systems is the management of their information sources. Basically, the success of a mashup depends on the one hand on the information and services available. On the other hand, it depends on the designer's creativity to mix them and obtain an added-value.

This work is focused on the first issue. It describes a middleware intended to provide information and services to mashup systems from a mobile and ubiquitous learning ecosystem. This ecosystem consists of several mobile devices that interact with different learning applications. These applications are personalized with the students' geographic location and profile (contextual information).

Furthermore, the other main focus of the paper is the use of a reconfigurable plug-and-play architecture that supports the integration of different services with the mobile clients.

The paper might be of interest for those involved in the development of mashup systems based on mobile devices, context-aware systems, and e-learning tools, as a vehicle to improve the educational experience through customized services and information to both students and academic staff.

II. OBJECTIVES

The main objective of the paper is the description of a middleware aimed to support the development of mobile applications through an simplified API. This API includes the following characteristics:

- Support for location-based technologies, such as GPS, RFID and / or Wi-Fi. Although it is also designed to support the integration of others.
- Context-awareness. The capability to manage user's information, e.g. profile, location, or history. This is called the user's context and is used to personalize the services provided to him/her. This context is retrieved in a transparent and automatic way, with no interaction from the user.
- Access to services provided by external e-learning applications, such as a Learning Management Systems (LMS). These services will provide communication and collaboration capabilities.
- A common Interface to access the contextual information and services. The middleware acts as a homogenizer layer that encapsulates different contextual information and services to create mobile applications in a simplified way.
- The integration of the external services into the middleware is achieved through the use of a reconfigurable plug-and-play architecture based on Web Services.

III. USE OF MASHUPS IN EDUCATION

The described system is intended to improve the user experience through the use of personalized services on mobile devices. Mobile systems that are used to improve the educational environment are called mobile learning or m-learning applications.

When we talk about m-learning, it is possible to assume that m-Learning encapsulates the many different activities in an educational environment with a mobile device. This trend basically moves the traditional learning, with a teacher delivering a class in a classroom, towards the mobile world. Many examples can be found in literature, for example in [1] an application that reads a chapter of a book on a mobile device is illustrated in a karaoke-style; or the MPSS project [2], where students must follow some courses and are evaluated through some test exercises using the mobile device.

On the other hand, there are other m-learning projects oriented towards outdoor learning. This is a totally different

point of view, because in these projects the idea is not to try to apply the old methodology into the new technologies, but to develop new environments where mobile devices offer an added value to students' education.

These environments do not try to supplement a particular class. They aim to complement the traditional methods and further formalise education; they open a new range of possibilities to students. For example, Bouvin [3] worked on a prototype where learners should explore and document parts of a city for later presentation in plenum.

Other projects more oriented to context-awareness such as Bomsdorf [4] allow learning materials to be selected depending on a given situation – it takes into account learner profiles such as their location, time available for learning, concentration level and frequency of disruptions. Similarly, a context-aware mechanism has been developed by Bouzeghoub et al. [5] which takes into account time, place, user knowledge, user activity, user environment and device capacity for adaptation to the user.

After many years of research in this field, our point of view is that m-Learning must complement both e-learning and traditional learning. In our opinion, each methodology has its own place. For that reason, we work with two kinds of mobile applications:

- Learning on the move: Devoted to education out of class, not replacing the face-to-face class, but complementing it, and providing different ways of learning in the field. In this line, m-learning seems to cater for certain specialties more than others such as agronomy, biology, geology, archaeology, and tourism, etc.
- Collaboration: Development of collaborative exercises together with other students.

Both kinds of applications perfectly match mashup systems, especially those based on geographical information services and other contextual information. Thus, mashups can create very valuable applications in the education area. They can combine learners' contextual information with other services intended to provide information, collaboration, communication, or just group-work. As a result, mashups are a very good tool to improve the care provided to learners by offering more personalized services.

IV. THE SERVICE-ORIENTED APPROACH

The Service-Oriented methodology that we are using in our development is based on the idea of having different services available on the Internet (on the Cloud) that can be used and integrated regardless of physical location. These services will come mainly from e-learning applications, as we work in the education environment. As examples of services provided by e-learning applications and used in our middleware we can find: forum rooms and chat from LMS, Wikis, Blogs, collaboration tools, etc.

This service-oriented trend is based on the concept of modelling the educative services with the objective of providing better interoperability capabilities in different levels: service-service, platform-platform, and service-platform. Equally it is based on the encapsulation of learning

objects services. The aim is to develop autonomous and self-contained services that can be easily integrated in different environments such as Learning Management Systems and mobile learning environments.

In addition, there is another surrounding concept to the service-oriented methodology within educative services: Digital Educative Ecosystems [6]. This concept is based on the idea of creating environments made of different integrated systems (mobile clients, applications, services and other tools) devoted to improve the learning experience by supporting communication and collaboration. One of the bases of this concept is the use of SOA (Service-oriented architecture) technologies such as SOAP [7], Web Services [8], ESB (Enterprise Service Bus) [9] or REST [10].

V. M2LEARN MIDDLEWARE DESCRIPTION

The successful implantation of education services on mobile devices requires the development of applications that are able to provide the needed support to the mobile learning experience. Thus, authors have committed to create a middleware devoted to support the development of new generation mobile and ubiquitous learning applications. Its design has taken into account the integration of the existing e-learning resources. The reason is that e-learning platforms are currently the centre of the on-line and blended education. These platforms are a repository of both content and services that has been developed and refined over multiple years. That is why mobile applications should not be designed independently of the already developed e-learning standards and platforms, without taking advantage of all the existing resources.

Within the mobile environment, existing e-learning resources can be very useful to get a better personalization of the services provided to students. For example, using the credentials (login and password) it is possible to retrieve information about student's educational environment such as his/her degree or subjects. This information will be provided to higher level applications to personalize the services offered to the student. In fact, there is a lot of contextual information about the student that can be used to improve the learning experience: for example, personalized information services or search results depending on the user's geographical location.

M2Learn middleware interacts with the university LMS through a set of Web Services that provide the interfaces required to implement mobile device functionalities e.g. access to forum rooms, content or FAQs. In addition, there are other advantages that relate to the re-use of services instead of creating them again: authentication, tracking activities, evaluation, content, etc. LMS's can also provide very useful contextual information.

On the other hand, social interaction and collaboration is one of the key issues for modern learning. It has been widely integrated in e-learning environments and must be moved into the mobile space.

The M2Learn project has as its main goal the simplification of the mobile learning developments. There

are four main middleware modules (Figure 1):

- Location Manager - Manages the information from the location sensors (e.g. GPS, RFID, Wi-Fi) to provide contextual information about the user's location. The system has been designed using open interfaces in order to simplify the addition of other technologies (e.g. new location methods such as cell towers or Bluetooth) [11].
- Profile Manager - Manages a user's personal information, such as preferences, contact information, or credentials. This module also manages other interesting information about the device: operating system, Internet browser version, etc.
- Post-processing manager - Processes the contextual information provided by the location and profile managers to access other services, e.g. information in a LMS, e-mail, etc to get more contextual information.
- Adaptation layer - This is the most important layer in the architecture because offers an interface (API) to top level applications.

These modules make up the M2Learn Unit. It is the API that higher level applications use to access the contextual information and personalized services.

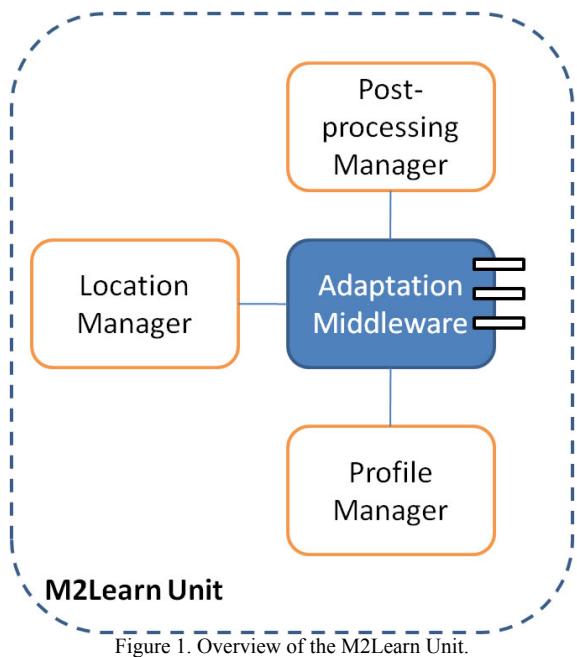


Figure 1. Overview of the M2Learn Unit.

This unit uses external e-learning services and resources through the reconfigurable plug-and-play framework [12]. This architecture will be described in more detail in the following section.

This interaction is achieved through a set of Web Services. They provide an interface to implement mobile device functionality, such as for example, access to forum rooms, content, or FAQs. In addition, there are other advantages related to the re-use of services instead of creating them again: authentication, tracking activities, evaluation, content, etc. The LMS can also provide very useful contextual information.

However, a mobile ecosystem cannot be conceived with one single system. That is why our scenario envisages the coexistence of several M2Learn Units that communicate and collaborate simultaneously.

These features can be accomplished using the existing e-learning services. However, the middleware also provides other modules that considerably extend this functionality: The Context-Hub module receives the user's contextual information and allows its distribution (Figure 2). Using the Context-Hub module users can discover information about other surrounding users. In addition, this module is also fundamental to support the development of mashup systems. It provides contextual information of all the mashup system users which can merge location, preferences, history and other services.

The interconnection between M2Learn Units, Context-Hubs, and e-learning resources and mashup systems is carried out through the Reconfigurable Plug and Play Framework. Therefore, these modules interact with this framework any time they need to connect to any of the available services.

The Reconfigurable Plug and Play Framework can mashup the required services. As Figure 2 shows, the mashup can be accessed using two methods:

- Through the Reconfigurable Plug-and-Play Framework, if it has been developed over the M2Learn Unit, it means, using the M2Learn API.
- Directly from the mobile device's browser, if it only uses the information from the Context-Hub. For example, as a Web application that receives a XML file from the Context Hub.

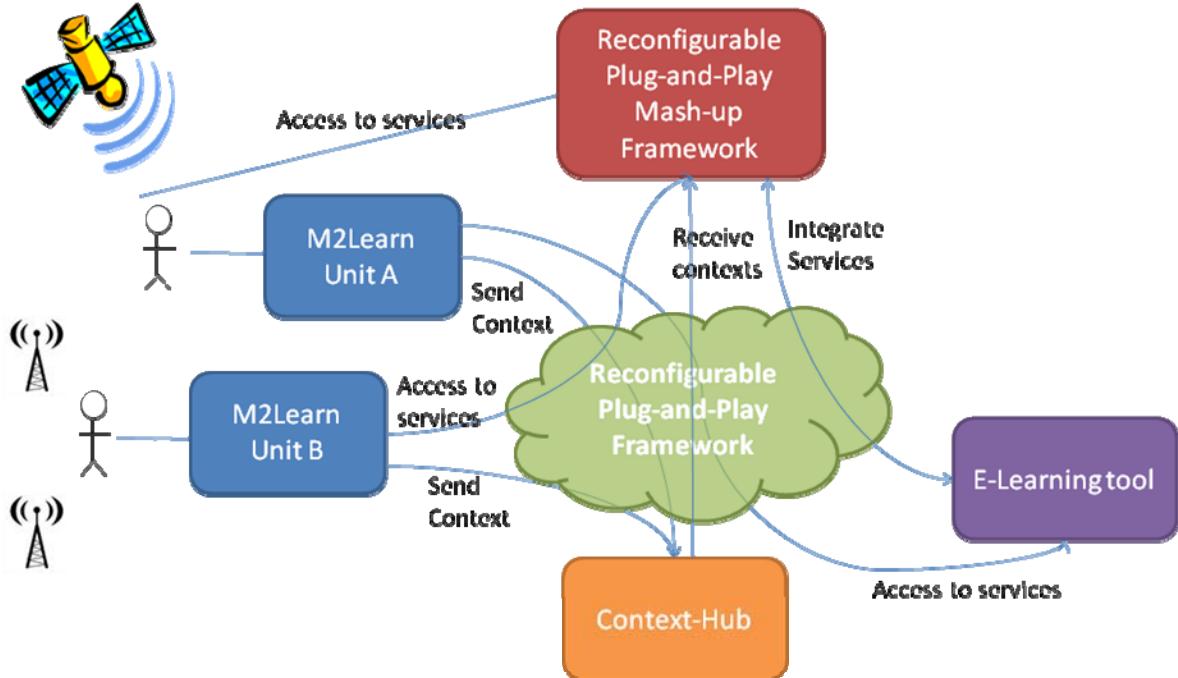


Figure 2. Interconnection of Modules through the Reconfigurable Plug-and-Play Framework.

VI. INTEROPERABILITY OF SERVICES THROUGH A RECONFIGURABLE PLUG AND PLAY ARCHITECTURE

The ability of systems to adapt or be adapted to disparate enterprise requirements and environmental changes is referred to as their flexibility. A flexible system is needed due to the requirement for a system to be deployed in converged enterprise (diversity) and to be flexible to evolving requirements (uncertainty). Versatile systems exhibit generic and function rich properties as a response to rapid and frequent development, maintenance, customization and evolution.

Service-based software engineering is a way of raising the level of abstraction for software development so that software can be built by easily reusing previously designed, implemented and refined converged services. Composite architectures that incorporate enterprise services are formed using a Service-Oriented Architecture (SOA) as a standardized way of connecting loosely-coupled systems.

Re-configurable service oriented architectures promote simplified software composition. They can be used to mashup enterprise business applications.

The architecture used to integrate the external services with our middleware as well as the mash up system is a component-based framework developed by Talevski [12]. It is a reconfigurable plug and play framework that is aimed at eased software composition, customisation, maintenance, and evolution. This architecture consists of four main modules:

- Software platform and framework
- Reconfigurable database
- User management system

- Tailoring tools

Using a reconfigurable plug and play framework as a basis for the creation and modification of software, it is possible to construct, customize, integrate and evolve solutions in a straightforward way.

The framework allows the solution to easily integrate existing applications and newly implemented components as services. The architecture defines the following mashup interface (Figure 3);

- Discovery - Plugin discovery is used to find, identify, describe and use available plugins. In order to locate plugins, a broadcast request for a plugin lookup service is sent. Each plugin responds to this request with a lookup proxy.
- Query - It is possible to query the plugin lookup service for available attributes and operations.
- Identification - The plugin lookup service is used to define plugin characteristics.
- Invocation - When a plugin is selected it dynamically binds and invokes its entrypoint. Numerous attributes and operations may be available. Each attribute and operation defines its prerequisites and post execution conditions.

One of the most relevant features of this architecture for its integration with M2Learn is its tailoring function. Tailorable software allows generic systems to satisfy the specialised, rapidly changing, unclear and/or evolving changes in system requirements. It provides a means for the straightforward creation and modification of software at runtime based on multiple levels of detail and complexity.

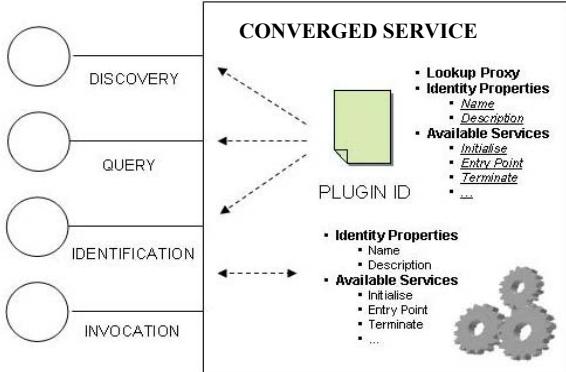


Figure 3: Converged Service

According with Morsch's classification [13], there are three tailoring activities where the system can be modified dynamically [14]:

- Customisation – It refers to changes to the user interface, so it is not relevant in the integration with the M2Learn middleware.
- Integration – Enterprise business processes tend to rapidly change in order to meet market needs. Consequently, their software systems must also change. The integration level of tailoring refers to the changes in components, compositions, interconnections and their configurations.
- Extension – A system evolves as new requirements are introduced. This may include the introduction of new components and configurations. The extension level of tailoring refers to the addition of new application components and configurations.

These features are particularly important, as the mobile environment may change frequently. Therefore, this framework provides the flexibility that a mobile ecosystem requires to dynamically evolve. This could even support the addition or modification of new services at runtime.

VII. APPLICATION TO MASHUPS

Regarding the Context Hub, it has been developed to allow all users of the environment to send their contexts to others [10] with 4 operation modes that offer different information depending on the requirement:

- Send own context.
- Receive the context of a particular user.
- Receive the context of a particular group.
- Receive the context of all the users in the environment.

This process is done every several seconds, depending on the configuration of each M2Learn Unit. It allows the user to have a very wide view of the other user's actions in the same environment. Thus, it offers several receiving modes: user, group and all. The first one is oriented to monitor the movements and status of a particular user.

The second one is intended to monitor the status and movements of all the users of a particular group. And the last mode allows monitoring of all the users located in the same area. The management of these modes depends on the privileges that the user has to access a certain group or user's information.

In addition, as it has been previously underlined, other external applications and services can receive this contextual information, as is the case of Mashups. They would simply request information from the Context-Hub, if they are allowed to do so, and they will receive an XML file with the users' contextual information. The Context-Hub can provide geographical location of a user, a group, or other information, such as preferences, history, etc.

VIII. CONCLUSIONS

The work introduced in this paper describes a powerful architecture that supports the development of different kinds of applications. This paper especially focuses on the creation of mashups, although it can be also used to implement any kind of context-aware application.

One of the contributions of this middleware is the homogenization of contextual information and services that simplify the development of advance tools with a considerably reduced effort. For example, users will be able to create a mashup system using the location information instead of learning how the NMEA protocol of the GPS works on a mobile device; or communicating through a serial port with an RFID controller to read the information from an RFID tag, without understanding the RFID commands or the data organization inside the tag. Or, they can create a mobile blog using the services provided by an e-learning platform, but they will not need to create any web services in the platform language or understand how its database is structured. They just use the simple interface with information and services provided by the M2Learn Middleware.

The second main contribution is provided by the Reconfigurable Plug-and-Play Framework. It supports the development of a flexible service-based architecture that perfectly matches with the mobile computing requirements that the M2Learn project aims to develop.

Finally, from a more general perspective, this middleware opens the door for the development of mobile and ubiquitous ecosystems in the educational environment. These systems consist of context-aware applications (with the added-value of geographic location), mobile clients, and external services (in charge of providing communication and collaboration capabilities).

The development of these kinds of architectures makes a step forward in the final arrival of ubiquitous and mobile technologies in society. And in particular, the M2Learn project positively impacts the educational environment.

ACKNOWLEDGMENT

Authors would like to acknowledge the Spanish Science and Education Ministry and the Spanish National Plan I+D+I the support for this paper with the project TIN2008-

06083-C03/TSI “s-Labs – Integration of Open Services for Remote and Virtual Labs”; and the European Union for the Leonardo Project 142788-2008-BG-LEONARDO-LMP “mPSS – mobile Performance Support for Vocational Education and Training Project” and the ERASMUS Project 141944-LLP-1-2008-1-ES-ERASMUS-ECDSP “IPLECS Internet-based Performance-centred Learning Environment for Curriculum Support”. And also the CYTED-508AC0341 “SOLITE- SOFTWARE LIBRE EN TELEFORMACIÓN” and the Complimentary Action TSI2007-31091-E “Objetos educativos reutilizables (para el EEES en las especialidades en las especialidades de las tecnologías de la información y las comunicaciones)” support.

REFERENCES

- [1] Almirall, M., Rivera, J.M., “Automatic Mobile Learning Contents,” *Proceedings on the 2009 IADIS Mobile Learning Conference*, Jan. 2009, Barcelona (Spain).
- [2] Mobile Performance Support System (MPSS): <http://mpss.ath.cx/> Queried on: 3th of April, 2009.
- [3] Bouvin, N., Brodersen C., Hansen, F., Iversen, O., and Nørregaard P., “Tools of Contextualization: Extending the Classroom to the Field,” *Proceedings of the 2005 conference on Interaction design and children*. Boulder, Colorado, Pages: 24 – 31, 2005.
- [4] Bomsdorf, B. (2005) “Adaptation of Learning Spaces: Supporting Ubiquitous Learning in Higher Distance Education”, *Dagstuhl Seminar Proceedings 05181 Mobile Computing and Ambient Intelligence: The Challenge of Multimedia*.
- [5] Bouzeghoub, A., Do, K. and Lecocq, C., “Contextual Adaptation of Learning Resources”, *IADIS International Conference Mobile Learning*, pp. 41-48, 2007.
- [6] Fontela, J., et al, A SOA Architecture for e-Learning Systems through the Seamless Merging of Web Services. Towards an Ecosystem for e-Learning Widgets. *Journal of Systems and Software*, Elsevier, June 2009.
- [7] Bell, Michael, B., "Introduction to Service-Oriented Modeling". *Service-Oriented Modeling: Service Analysis, Design, and Architecture*. Wiley & Sons. pp. 3. ISBN 978-0-470-14111-3, 2008.
- [8] Chappell, D., “Enterprise Service Bus”. O'Reilly Media, Inc., 2004. ISBN:0596006756
- [9] Prescod, P., "Second Generation Web Services", O'Reilly's XML.com Website. Last accessed on February 06, 2002, <http://webservices.xml.com/pub/a/ws/2002/02/06/rest.html>
- [10] Brogden, W., “REST versus SOAP – the REST story”, Website. Last accessed on November 20, 2009 http://searchwebservices.techtarget.com/tip/0,289483,sid26_gci1227190,00.html
- [11] Martín, S., Bravo, J., Hervás, R., Sancristobal, E., Gil, R., Díaz, G., Losada, P., Castro, M. y Peire, J., “Location-based Services for Mobile Devices at University”, *IMCL, International Conference on Interactive Mobile and Computer Aided Learning*, pp. 1-7, April 16 - 18 2008, Amman (Jordan).
- [12] Talevski, A., Chang, E., and Dillon, T.S., “Reconfigurable Web Service Integration in the Extended Logistics Enterprise”, *IEEE Transaction on Industrial Informatics*. May 2005, Vol. 1, Number 2. ISSN 1551-32-03.
- [13] Morch, A., “Three Levels of End-User Tailoring: Customisation, Integration, and Extension”, *Computers and Design in Context*, The MIT Press, pp: 51-76, Cambridge, Massachusetts, 1997.
- [14] Martín, S., Sancristobal, E., Díaz, G., Gil, R., Castro, M., and Peire, J., A framework based on location technologies to build mobile context-aware applications. *Proceeding on 1 Simposio Sobre Computación Ubicua e Inteligencia Ambiental*, September 2005, Granada (Spain), pp 389-395. Thomson, ISBN: 84-9732-442-0.