

# EXPEL Protocol Based Architecture for Cooperative E-Learning

<https://doi.org/10.3991/ijet.v11i10.5543>

Karima Aissaoui, El Hassane Ettifouri, and Mostafa Azizi  
Mohammed First University, Oujda, Morocco

**Abstract**—Using E-learning platforms in learning process become a common phenomenon because of the large use of Information and Communication Technologies (ICT) in all fields. A huge number of e-learning platforms were developed what makes their list richer and bigger. Most of teachers have the possibility to choose the suitable platform for their needs. Consequently, we may find more than one platform used in the same institution, and thus students have to connect independently on each system. This diversity in use of platforms makes the learning process for students more difficult which discourage them to follow online courses. In this paper, we propose a new architecture to solve this problem by unifying access and managing cooperation between platforms used in the same institution.

**Index Terms**—E-learning, Platforms, Cooperative systems, Access Control.

## I. INTRODUCTION

E-learning is defined as the use of electronic tools in delivering instructions including CD-ROM, Internet, Intranet, Extranet, audio, video, etc [1]. Many institutions around the world integrated this way of learning in their programs in order to improve the quality of learning/teaching. This electronic approach of teaching uses tools and methods to offer an environment where students and teachers can collaborate not just in ordinary classrooms, but also in different contexts. Using E-learning systems, learning content is managed and delivered online [2]. However, we find that many systems and platforms were developed and proposed which makes the choice of a suitable platform personal and depends on the needs of users. As a consequence, in an institution where more than one system is used, students must be connected in all platforms used by their teachers independently. Furthermore, they must be familiarized with all platforms. On the other hand, if two teachers do not use the same platform, they will not have a collaborative space where they can exchange their courses and groups of students. Our proposed architecture is a real solution for those problems. First, we will offer to students a common system which will avoid them to connect independently on every platform. In addition to this, using our architecture, teachers will be able to collaborate even if they do not use the same platform.

More and more the list of e-learning systems offered in the market becomes longer, more the user demands a highest quality and more efficient systems that meet the needs of supporting agile, consistent and collaborative learning. However, platforms offered, do not support collaboration between different systems.

In great majority of institutions, we find more than one platform is used. For example, in our university (University Mohammed First in Oujda- Morocco) [3], two main platforms are used: Moodle [4] and Claroline [5]. Furthermore, a teacher can sometimes use both systems at the same time. Behind this situation, students are obliged to be connected separately in all platforms used by their teachers to have access to resources of every platform.

With the large use of those systems, the needs of improvement of the access mechanism to e-learning platforms have appeared. First, collaboration between teachers is necessary in order to work together. For this goal, they need a common system where they can exchange information and groups of students. In addition to this, when more than one platform is used by teachers, students are front of redundant and nonflexible systems. Moreover, students of teachers using more than one platform must be connected independently in all systems used by their teachers. This task becomes redundant and discourages them to profit from the benefits of such learning method. Our proposed architecture comes as a real solution to those problems. In the next section we will present some related works relative to collaborative e-learning.

The remaining of this paper is organized as follows: in the second section, we give the state of art of collaborating e-learning. After this, we present our architecture. In the fourth section we detail deeply the different components of the proposed architecture, we explain how communication between them is done and we illustrate implementation of the system. Finally, we discuss our solution and we give some perspectives and future works.

## II. RELATED WORKS

### A. Cooperative e-learning system BluES'n

“Every user should have access to all functionalities provided by the environment” was the main principle from which developers of the cooperative e-learning system BluES'n [6] have started. In this system, the concept of real classrooms and learning areas was replaced by the notion of workspace. They defined two types of workspaces: Shared which provide collaborative and communication environment, and Personal workspace which is special for every user. These workspaces are created and configured by every user according to its requirements. Functional modules, contents and users and their roles are the main characteristics of an individual workspace. Using those modules, users get access to educational functions and tools that allow collaborative exchange of contents and documents, communication and interactions between users. Thus, collaborative workspaces need roles in order to manage responsibilities of different users. Here, the

authors talk about two types of roles: administrative and functional. Access rights to special resources in a workspace are described in administrative roles which are defined as: the owner of the workspace, participants and guests. On the other hand, we find functional roles such as: tutors, teachers and moderators.

Even if the BluES'n system provides a collaborative e-learning environment to its users according to access rights and roles specifications determined, which could ensure the secure policy, this system still do not offer the required effectiveness and the collaboration level demanded.

### B. Unified access using LDAP Directory

In a previous work [7], we have proposed architecture in order to unify access to different platforms used by different teachers of the same group of students. This architecture is based on LDAP (Lightweight Directory Access Protocol) directory [8]. Using the system proposed, the administrator can choose the platform to add via a simple web interface. Then, users of the different platforms installed will be stocked in the LDAP directory of the common system. Consequently, the access to all platforms installed will be unified because one directory is shared between them. So, students will have the ability to connect just once on all platforms and to switch between these platforms using a web interface.

The proposed system can collect users' data from all platforms used and put it in an LDAP directory. This specification makes easier the management of users of e-learning platforms in an institution, but it has also some disadvantages. First, data stocked in an e-learning system can be huge in institutions where students are numerous. This constraint cannot be managed via a simple web interface. Also, when the number of platforms installed in the system becomes bigger, the management of access rights and mapping between all platforms becomes more difficult. For those reasons, we propose in the following an architecture that comes as a real solution to improve the cooperation between e-learning systems.

### C. Cooperative e-learning using SCHOM tool

SCHOM (SCHolar Messaging) is a complement that comes to improve e-learning systems [9]. It allows its users in university context to send messages using different types of devices (Mobile devices, computers running on different operating systems) and multiple communication channels (Social networks, instant mobile messaging, email, etc.). Using SCHOM tool, collaborative e-learning will be easier for workgroups. The messages that can be sent using this tool are classified according their persistence. Also, users are managed via an Access Control List (ACL) [10], what ensures the security of access.

On one hand, the proposed system allows its users to have an interoperable tool that they can use in order to improve communication between them even if they work on different devices or they use different communication channels. On the other hand, SCHOM does not consider teachers who are habitual to use existing e-learning systems. That means that the problem of cooperation between e-learning systems with existing data was not really solved. In this paper, we will try to give the appropriate solution and to help users to have a real collaborative system.

## III. PROPOSED ARCHITECTURE

### A. Global view of the proposed architecture

Users of our systems are classified on two categories: administrators of the system and final users.

- Administrators are those who are responsible of the technique side of the platform. They install the e-learning platform, create accounts and manage users.
- Final users of the platform: They are students and teachers. Sometimes we find that teachers can be administrator and final user at the same time.

Our system is composed by three main elements:

- Plug-in to install in the existing e-learning platform
- The common platform or the web interface from which the final user can connect.

In the next sub sections, we will explain all those elements on details.

The figure 1 is a global view of the proposed system.

ELP (E-Learning Platform) is used for e-learning platform;

SPEL P is used for the Special Plug-in of the E-learning Platform ELP;

GPELP is used for the Global Plug-in of the E-learning Platform ELP;

GELP is used for the Global E-learning Platform;

EXPELP is used for the EXchange Protocol for E-learning Platforms.

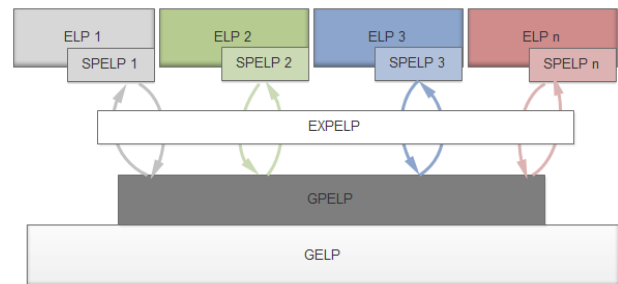


Figure 1. Global view of the proposed architecture

In the next sections, we will detail all elements of the proposed system.

### B. SPELP: The Special Plug-in

The Special Plug-in SPELP<sub>i</sub> is dedicated to the e-learning platform ELP<sub>i</sub>. This plug-in is composed from two parts:

The specific part which is special for the platform *i* (This part is specific for each e-learning platform and depends on its architecture and the language of development of ELP<sub>i</sub>). It communicates with the platform ELP<sub>i</sub> in order to obtain data. Here, we find two cases:

- Case A: The ELP<sub>i</sub> communicates those data using services.
- Case B: The ELP<sub>i</sub> has only the database, so the specific part interrogates it in order to obtain data.

The common part is the front interface. It is common between all plug-ins of all platforms. The figure 2 illustrates the structure of the SPELP.

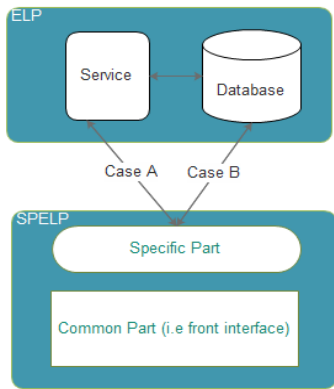


Figure 2. Structure of SPELP

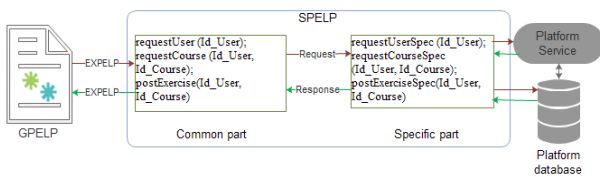


Figure 3. Communication between GPELP and SPELPi

The communication between the GPELP and the SPELPi is done using the EXPELP protocol. The Common part or the front interface sends requests to the specific part through functions that are already defined (we will define those functions in the next sub-section). The specific part sends these requests to the service or the database (See section V for more details).

Some functions that are defined in the common interface are (but not limited to just those functions):

- Request User (Id\_User): this function is defined in the common part of SPELP and it is used in order to request a user by its ID.
- Request Course (Id\_User, Id\_Course): this function requests a course.
- Post Exercise (Id\_User, Id\_Course): this function is called when a user wants to post a course.
- Request UserSpec (Id\_User): this function is a response from the specific part to the common part. It is used in order to answer the request request User (Id\_User).

The figure 3 explains how the GPELP communicates with SPELPi.

The Communication between the common interface GPELP and the specific plug-in SPELP is done using a new protocol that we created. We will explain its architecture and its specifications in the next section.

### C. EXPELP: Protocol Of Exchange

#### 1) EXPELP's architecture

Our contribution in this article makes cooperation between different e-learning platforms possible. Our proposed system is based on plug-ins installed on both platforms ELPi and the global platform GPELP. The communication between those plug-ins is done using EXPEL protocol. We have created this protocol based on JSON (JavaScript Object Notation) [11] for data representation, REST (Representational State Transfer) [12] and HTTP (Hypertext Transfer Protocol) [13] for transfer of data.

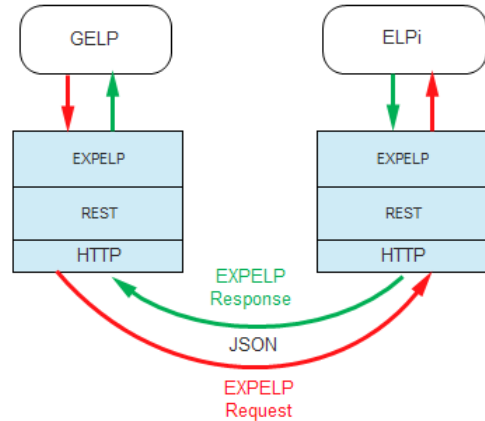


Figure 4. Layers of EXPELP

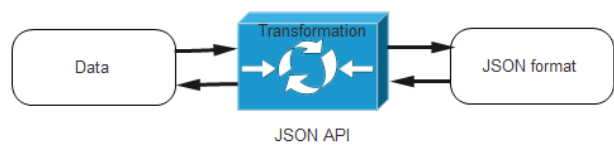


Figure 5. Bidirectional transformation between data and JSON format

The figure 4 shows us the structure of the EXPEL protocol. It is composed by three layers. The principle of communication is requests/responses.

#### 2) The choice of technologies

The question that has appeared to choose the interchange data was: between XML (Extensible Markup Language) and JSON which is the more suitable for our case? To answer this question, we have looked for a comparison between these two tools. According to [14], XML [15] is used primary in Remote Procedure Calls (RPC) [16] and object serialization for transfer of data between applications. In our case, we have chosen JSON because of the multiple requests and data that are exchanged between GELP and ELPi which makes of the speed of data exchange a primordial factor. Data transformation between the platform and JSON format is done in a bidirectional way. The figure 5 illustrates this transformation.

Representational State Transfer is an architecture style for designing networked applications [12]. It can be considered as a lightweight alternative to Web Services and RPC. Applications based on this architecture use HTTP requests in order to do CRUD operations (Create, Read, Update and Delete). We have chosen this tool for several reasons. First, REST is Platform independent. That means that there is no care if the client and the server do not run on the same platform. Also, it is independent of the language which allows two different systems developed using different languages to communicate without any problem. In addition, REST uses HTTP which makes this tool based on this standard and gives it a powerful character.

Briefly, in order to obtain a high level of abstraction, and consequently a very interoperable architecture, we have decided to use REST.

#### 3) Communication Between GELP and ELPi

The communication between the general platform and the specific one is done according to the Client/Server architecture. First, the user is logged using GELP interface which sends a request to the LDAP server in order to

obtain a key for all next communications between GELP and ELP special to this user. Then, if the user has an available account on this directory, a response will be addressed to the GELP. After this, a request will be sent to the ELP which will also check the availability of this key on the LDAP directory. This key is used as a security mechanism. When all those operations are done successfully, the communication between GELP and ELPi can be done. The figure 6 explains how all those operations are done.

IV. DISCUSSION

Most of institutions try to improve the learning process using different methods. One of those methods is the integration of new e-learning systems in the learning process. Behind this situation, the administration must update the system in order to complete the cooperation between all used platforms in the same institution.

Two simple steps must be followed to add the new platform to our proposed system. First, the specific part of the SPELP must be created. As said above, the SPELP plug-in is composed by two parts: public and specific. The public one is common and does not need any modification since it communicates with GPELP. Consequently, the administrator of the platform should implement only the specific part of the SPELP plug-in, which will be in direct communication with this new platform.

The second step is to add the URL of the new implemented SPELP in the general plug-in, in order to have a channel of communication between both plug-in.

The improvement of the learning process in institutions still one of research axes that knows the field of learning. For the same goal, most institutions offer to their students and teachers all the freedom in choosing the e-learning platform suitable for their needs. Our aim is to help those institutions to have a unified access system to all platforms used.

Using our proposed architecture, students will not be forced to access independently all platforms used by their teachers (for example one student needs to access Moodle used by teacher 1 and Claroline used by teacher 2). He/she (the student) must log in only one time using the GELP interface. Also, teachers will be able to collaborate between them without any need to use only one platform. In the side of administration, there is no doubt if a new platform is coming. Two simple steps must be done in order to add this new platform to the system.

The great contribution of this system is to help the student, the teacher and the administration to improve the context of using different platforms in the same institution. Thus, the proposed architecture will improve the learning process because the student will focus his/her efforts on the task of learning without the need to switch from a platform to another. Also, the user of our proposed system will not feel the background architecture since we use light and fast tools. In addition to this, the security of the system is also studied. We have implemented a solution that uses a security key in order to exchange data.

All these characteristics make our solution the perfect one to face problems of heterogeneity of using numerous platforms in the same institution.

A. Implementation of the proposed system

1) Application architecture

The architecture of system is illustrated in figure 7. Interaction between the user and system begins with the view layer. Here, the GELP constitutes the Man-Machine Interface [17] of the system. Authentication data of user are transmitted to the controller layer for verification. After this, EXPELP protocol is responsible for all interactions between specific e-learning platforms and our proposed system. This communication is explained above in the previous sections.

2) Use cases

In e-learning, students and teachers constitute the core of this type of systems. Starting from this, we focus our next diagrams and implementations on those two principal actors. The figure 8 illustrates the most important use cases according to UML notation [18] of students in e-learning systems.

As seen, the most important functionalities used by students in e-learning platform are relative to courses and exercises. The figure 9 schematizes the teacher's use cases. As in the case of students, only operations relative to courses and exercises are shown in this diagram.

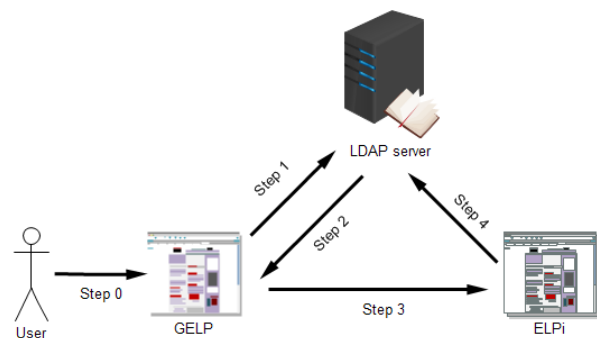


Figure 6. Scenario of communication between GELP and ELPi

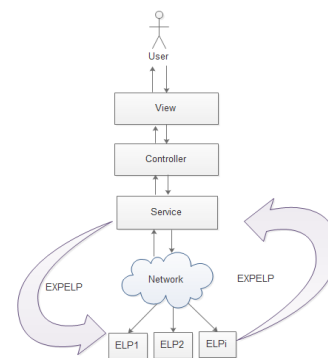


Figure 7. Application architecture of the proposed system



Figure 8. Student Use case

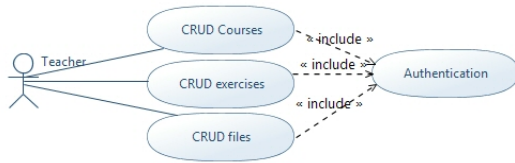


Figure 9. Teacher Use case

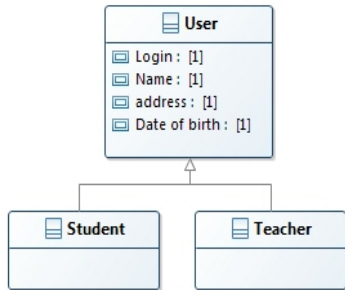


Figure 10. User class diagram

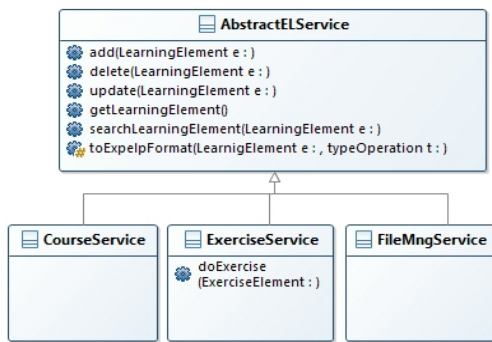


Figure 11. User class diagram

3) Class Diagrams

We focus in this sub section on class diagrams relative to two types of users (student and teacher) (figure 10), the layer service where different paramount functionalities (figure 11) and all components are modeled (figure 12).

As said above, the paramount functionalities are relative to courses and exercises. In the next figure (Figure 11), we include in the class diagram different methods allowing operations of interaction between users and the system. These operations include adding, deleting, updating and searching a learning resource (course, exercise or file). Another method is dedicated to transform a learning resource to EXPELP format in order to transmit it using this protocol.

The process of transforming a learning resource to EXPELP format is done according to the mechanism explained in the figure 12.

4) Sequence Diagrams

When the user requests a page (searchCourse for example), his/her login and password are used in order to generate the security key. If this operation is done successfully, the welcome page is generated and opened for the user who can choose the operation to execute. After this, the GELP interface sends a request using the EXPEL protocol demanding data in EXPEL format from the specific platform which also verifies the security key and finally the user is redirected to the page requested. The figure 13 illustrates this diagram.

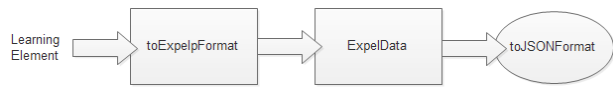


Figure 12. Process of transformation of learning elements to JSON format

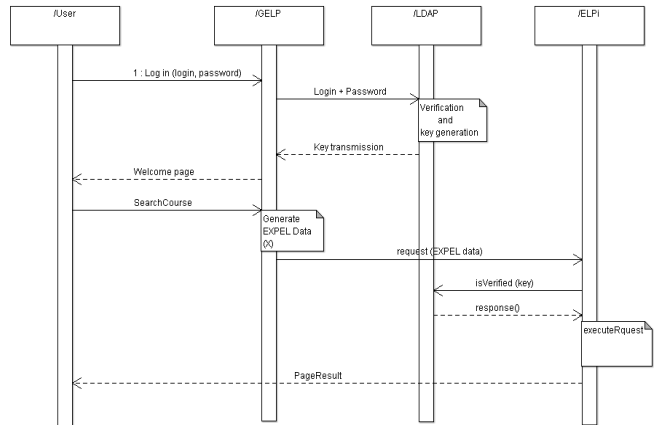


Figure 13. Sequence Diagram of requesting a functionality using EXPEL Protocol

V. CONCLUSION

In a situation where teachers use different platforms, the student needs to have access to all platforms used by his/her teachers independently. In order to help institutions to have a centralized system where all platforms used could collaborate, we have proposed a new architecture based on plug-ins and EXPEL Protocol. Using this system, students will not need to access to every platform independently. Moreover, cooperation between e-learning systems becomes possible which will improve the learning process and help the actors to work in a perfect context without the need to be limited in a unified platform.

In this paper, we have proposed a new architecture that can be applied not just in e-learning domain, but also in other fields where there is a need of unifying access and collaboration environment. Actually, we are working on the implementation, verification and validation including tests of this approach on some existing platforms in order to offer a complete system with a big interoperability, to make it more suitable even for possible new environments.

REFERENCES

- [1] Rosenberg, Marc Jeffrey. *E-learning: Strategies for delivering knowledge in the digital age*. McGraw-Hill, New York, Vol. 9, 2001.
- [2] O'Neil, Harold F., and Ray S. Perez. *Web-based learning: Theory, research, and practice*. Routledge, 2013.
- [3] Mohammed First University, <http://www.ump.ma/index.php/en/> [Accessed 23 May 2016].
- [4] Horvat, A., Dobrota, M., Krsmanovic, M., & Cudanov, M. *Student perception of Moodle learning management system: a satisfaction and significance analysis*. *Interactive Learning Environments*, 23(4), 515-527. 2015. <http://dx.doi.org/10.1080/10494820.2013.788033>
- [5] Lebrun, M., Docq, F., & Smidts, D. *Claroline, une plate-forme d'enseignement et d'apprentissage pour stimuler le développement pédagogique des enseignants et la qualité des enseignements: premières approches*. AIPU Seminar, Montpellier, May 2008.
- [6] Liesebach, Katja, and al. *Collaborative E-learning*. Digital privacy. Springer Berlin Heidelberg. 657-677. 2011.

PAPER  
EXPTEL PROTOCOL BASED ARCHITECTURE FOR COOPERATIVE E-LEARNING

- [7] Aissaoui, Karima, and Mostafa Azizi. *Taxonomy and Unified Access of E-Learning Platforms*. Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015. Springer International Publishing, pp 625-629, 2016.
- [8] Joshi, Chandrajit G., and Romil J. Shah. *Portable lightweight LDAP directory server and database*. U.S. Patent No. 9,032,193. 12 May 2015.
- [9] Berjón, Roberto, and al. *SCHOM. A tool for communication and collaborative e-learning*. Computers in Human Behavior, 2015, pp 1163-1171.
- [10] Jin, Xin, Ram Krishnan, and Ravi S. Sandhu. *A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC*. DBSec 12. pp 41-55. 2012. [http://dx.doi.org/10.1007/978-3-642-31540-4\\_4](http://dx.doi.org/10.1007/978-3-642-31540-4_4)
- [11] Shin, Sang. *Introduction to json (javascript object notation)*. Presentation [www.javapassion.com](http://www.javapassion.com) .2010.
- [12] Chen, Steven. *The concept of representational state transfer (rest)*. 2013.
- [13] Fielding, R., & Reschke, J. *Hypertext transfer protocol (HTTP/1.1): Message syntax and routing*. 2014.
- [14] Lin, B., Chen, Y., Chen, X., & Yu, Y. *Comparison between JSON and XML in Applications Based on AJAX*. In Computer Science & Service System (CSSS), International Conference on IEEE. pp 1174-1177. 2012. <http://dx.doi.org/10.1109/csss.2012.297>
- [15] Extensible Markup Language (xml) 1.0 (fourth edition). W3C, 2006. <http://www.w3.org/TR/xml/>. [Accessed 23 May 2016].
- [16] Merrick, Phillip, Stewart Allen, and Joseph Lapp. *XML remote procedure call (XML-RPC)*. U.S. Patent No. 7,028,312. 11 Apr. 2006.
- [17] Hu, M., & Kuang, Y. *Human-machine interface: Design principles of pagination navigation in web applications*. In Computer Science & Education (ICCSE), 2014, 9th International Conference on (pp. 1140-1143). IEEE.
- [18] Evans, A., France, R., Lano, K., & Rumpe, B. *Developing the UML as a formal modelling notation*. arXiv preprint arXiv:1409.6928. 2014.

AUTHORS

**Karima AISSAOUI** (aissaoui.karima@gmail.com) graduated from ENSAO (National Higher School for Applied Science, Oujda) with a degree of state engineer Software Quality in 2010. She is currently a Ph.D. candidate at MATSI-Laboratory/ Mohammed First University of Oujda under the supervision of Pr. Mostafa AZIZI. Her research activities focus on cooperation between e-learning platforms, and integrating services in m-learning. She has served as a reviewer for several international conferences and journals.

**El Hassan ETTIFOURI** (h.ettifouri@gmail.com) graduated from ENSAO (National Higher School for Applied Science, Oujda) with a degree of state engineer Software Quality in 2010. He is currently a Ph.D. candidate at LSEII-Laboratory/ Mohammed First University of Oujda.

**Mostafa AZIZI** (azizi.mos@ump.ma) Ph.D., Ing, Professor, MATSI-Laboratory, ESTO, University Med 1st, Oujda – Morocco. His researches focused on: Verification/CoVerification, Testing, Computer Security, Software Development, Hardware/Software Systems

Submitted 01 February 2016. Published as resubmitted by the authors 14 April 2016.