2016 30th International Conference on Advanced Information Networking and Applications Workshops

Towards the Effective Software Development of an eLearning Platform Featuring Learning Analytics and Gamification

David Gañán, Santi Caballé, Robert Clarisó, Jordi Conesa
Department of IT, Multimedia, and Telecommunication, Open University of Catalonia,
Barcelona, Spain
{dganan, scaballe, rclariso, jconesac}@uoc.edu

***Abstract*— This paper presents a first step into the construction of an innovative web-based eLearning platform featuring Learning Analytics and Gamification called ICT-FLAG. The main goal and key decision at this initial stage of the platform development is the search and selection processes of the most suitable software development framework to guide and support the effective construction of the platform. Despite the availability of many architecture and application frameworks for creating web-based systems in the market, only a few are potentially suitable for the specific goals of our eLearning platform. In this paper, we first review existing frameworks in terms of pros and cons considering the characteristics of the project. Then, the search's outcomes go into a selection methodology whose results are presented and discussed, leading us to make a final decision on the software framework that will eventually support an effective development of the ICT-FLAG platform. The research reported in this paper is currently undertaken within the project "Enhancing ICT education through Formative assessment, Learning Analytics and Gamification" (ICT-FLAG) funded by the Spanish Government.**

Keywords—eLearning platform, software development framework, , web technologies, JavaEE, Spring, OSGi.

I. Introduction

The objective of this study is to choose the most suitable software framework for the construction of an eLearning platform aiming to provide learning management systems (LMS) with innovative services in terms of learning analytics and gamification. The development of this platform is the main technological goal of a research project called ICT-FLAG currently undertaken in the context of university degrees in the area of Information and Communication Technologies (ICT). In this context, our ICT-FLAG platform aims to design and build eLearning tools and services that can benefit students, lecturers, managers and academic coordinators. This impact will be mainly achieved through three innovative pedagogical and technological axes:

- Formative assessment tools that can provide immediate feedback by means of automatic assessment [1][2].
- Learning analytics that monitor the activity and progress of the on-line teaching and learning processes supported by e-Learning systems and applications, combining this information with other sources of academic and historical information.
- Gamification as an incentive scheme in order to motivate students to practice more frequently and increase their engagement in the learning experience [3][4][-5].

In addition to dealing with multiple components and services, the ICT-FLAG platform needs to support a variety of virtual campus platforms, given that universities have a wide range of alternatives, from closed-source in-house solutions to open-source platforms, such as Moodle.

Considering the above goals of the ICT-FLAG project, a modular architecture with the following features is required:

- Customization. Easily selection of which components are deployed in each installation.
- Distribution. Enable the distribution of components along different installations.
- Extensibility. Enable the easy extension of an installation with new components.

This modular architecture also needs to consider some additional architectural and application features, required to ensure extensibility and the correct performance of the system, namely: dependency injection, pluggability, distribution of modules, events / messages between modules, web support (and especially REST), testing support, easy deployment and configuration, and portability.

The desired characteristics for a modular architecture are hard to obtain if they are developed from scratch. This is why an application framework could facilitate development by providing the above features directly within the framework.

As previously mentioned, the aim of this paper is to select a suitable software framework which helps us construct our ICT-FLAG platform. In the first place, a set of available software development technologies and frameworks will be listed, and the most suitable ones for the project will be filtered. Then, a selection methodology will be defined for determining the best option, consisting in the following tasks: (i) comparing the framework alternatives' features and (ii) constructing a prototype for comparison. Finally, for each selected framework, the construction of a prototype will provide the following metrics for each alternative: (i) the cost of construction of the prototype and (ii) the performance of the prototype.

The remainder of the paper is structured as follows: Section 2 reviews the different software infrastructures and other

978-1-5090-2461-2/16 $31.00 © 2016 IEEE
DOI 10.1109/WAINA.2016.7

technologies available, and makes an initial filter of the most suitable ones. Section 3 presents the selection and prototype assessment methodology and Section 4 presents the results of the selection process and providing a final decision. Finally, Section 5 concludes the study by summarizing the main ideas and outlining next steps on developing the ICT-FLAG platform.

II. BACKGROUND

This section reviews the different available software development technologies and frameworks as well as existing out-of-box products that can be used for the construction of the ICT-FLAG platform. The first subsection introduces different programming languages and frameworks, and discards some of them considering the project needs. The second subsection discusses the adoption of out-of-box products in order to construct our platform. Finally, the last subsection makes a deeper study of the technologies selected previously.

A. Programming languages and frameworks alternatives

The list of programming languages and frameworks that can be used to create enterprise web applications nowadays is very large. Hence we only considered the most mature, well-known and extensively used. The following list was elaborated by comparing different programming language popularity indexes, and job offering pages, such as TIOBE [5], BuiltWith [6], PYPL [7] or Stack Overflow Careers [8], and updating the list with personal knowledge and further searching:

- Java EE
- Java with Spring
- Java with OSGi
- ASP.NET
- Ruby on Rails
- Flask (Python)
- Django (Phyton)
- Express (Node.js)
- PHP

On the one hand, from the above list those frameworks based on the Java language (i.e. Java EE, Spring and OSGi) provide important benefits, such as (i) a well-known programming language, (ii) platform independence, (iii) a large community around them and (iv) wide adoption.

On the other hand, ASP.NET is a relevant alternative to Java in order to develop enterprise web applications, having many of the benefits of Java technologies. However, the full dependency on Windows platforms reduces a great deal its interoperability capabilities, which was the main reason why this technology was discarded.

The Ruby, Python and node.js alternatives are very popular nowadays but they are in general not intended for enterprise applications, and do not provide some features, such as inversion of control, that are needed for modularization. Hence they were also discarded for the construction of this platform.

Finally PHP-based solutions were also discarded because PHP is not a pure object-oriented software and does not facilitate sound programming practices or guidelines (e.g., layering, programming to an interface, encapsulation, etc.), which are important non-functional requirements in this system.

B. Out-of-box products

Another option to construct the ICT-FLAG system to use an out-of-box product which provides some required functionalities for the project, and adapt it to the specific system requirements.

There are many types of out-of-box products, but the most similar in functionalities to the project are those used to portal creation and content management. Virtual learning environments (VLE) products also fit the project requirements.

There are some examples of previous works that use portal and content management products to construct their systems, such as Liferay [9][10][11] or Microsoft SharePoint [12][13]. However, it is more usual to find examples using VLE systems, such as Moodle , Sakai or Blackboard (e.g., [14][15] and [16] just for listing some of them) and even MOOC (Massive Online Open Course) management systems, such as OpenEdx, LearnDash or OpenMooc [17].

The main benefit of using an existing out-of-box product to build a system is that many of the features of the system are already developed, or in general they are easy to adapt to the project requirements. However, we found some problems of using an out-of-box product for the ICT-FLAG project construction, specifically:

- They include many features that are not required for the project and in most cases cannot be disabled.
- They need to be adapted or extended to fit the concrete requirements of the project.
- Certain important features, such as security and the institution model are not easily adaptable to the needs of the project.

In conclusion, we decided not to use an out-of-box solution so as to have more flexibility to meet the ICT-FLAG requirements, thus having as few dependencies on third-party products as possible.

C. Study of pre-selected alternatives

From the wide spectrum of web development technologies presented above, we discarded those that do not fit the modularity and interoperability requirements of the system to be created. As a result, three architecture and application frameworks based on Java (OSGi, Spring and Java EE) remain as suitable alternatives. In this subsection, these three technologies are presented with more detail and their most important benefits and drawbacks are highlighted.

C.1. OSGi (Open Services Gateway Initiative)

OSGi is a set of specifications [18] that define a dynamic component system for Java, where applications are composed of many reusable components or modules called *bundles* ([19] chapter 2) which hide their internal implementation from other components while communicating with them through services.

Each module registers its services into an OSGi *container* who manages the lifecycle of the module ([19] chapter 3) and instantiates the correct services when needed.

There are several implementations of the OSGi framework for different purposes; the most well-known are Apache Felix, Eclipse Equinox and Knopflerfish. On the other hand an OSGI runtime is needed for the deployment of OSGi applications; the most well-known ones are Apache Karaf, Eclipse Virgo and Apache Geronimo.

There are some works using OSGi as their base framework, such as [20] where a system providing onboard services for road vehicles based on OSGi is presented, and [21],[22] where different approaches to domotic systems based on OSGi are discussed. However, to the best of our knowledge, there are no web-based eLearning systems constructed with OSGi.

The main benefits of using OSGi specification as the application architecture are:

- Supports modularity by design.
- Customization. With OSGi the developer can decide which services installs in each deployment.
- Enables dynamic updates on the fly, without having to restart the server.
- Supports versioning, so different versions of the same service can coexist in the same installation.

Almost all the works analyzed focus on the benefits of the OSGi framework. However, the use of OSGi technology also has some drawbacks:

- Tendency to over-modularization.
- Steep learning curve.
- Great dependency from the chosen implementation in terms of configuration.
- A lot of theoretical documentation, but almost all real tutorials are outdated and the OSGi developer community is not big enough to provide good support.
- Difficult to adopt for enterprise application development [23].

C.2. Spring

Spring is an application development framework which includes an inversion of control container. It supports other functionalities, such as: (i) Aspect-oriented programming, (ii) Authentication and authorization, (iii) Data access, (v) Model–view–controller, etc.

To the best of our knowledge, there are very few updated works directly related with Spring technology, such as [24] where Spring is presented as a lightweight alternative to Java 2 EE. Moreover, the specific use of Spring to build eLearning systems is not reported in the literature.

The Spring framework presents the following benefits:

- Highly adopted, with a big support community.
- Runs on a servlet container like Tomcat (not requiring a full JEE server).

Nevertheless, Spring also presents some drawbacks:

- Big size and steep learning curve.
- Complex, with lots of XML files for configuration.

C.3. Java EE

Java EE is the standard in the Java world to create enterprise applications, and includes support for modularity and dependency injection among other enterprise application characteristics. Some of the most important technologies and APIs defined by the Java EE specification [25] are: CDI (Contexts and Dependency Injection); JPA (Java Persistence API); JMS (Java Messaging Service); EJB (Enterprise Java Beans); Servlets; JSF (Java Server Faces); and JAX-RS (Java API for RESTful Web Services).

JAVA EE applications run inside *containers* called *application servers*, which provide an execution environment and a concrete implementation of the JAVA EE specifications. Examples of *application servers* are: GlassFish (reference implementation), JBoss EAP, WildFly, Apache TomEE, WebLogic (Oracle) or WebSphere (IBM). There are also *servlet containers*, which implement only the JSP/Servlet part of the huge Java EE API, such as Tomcat or Jetty.

There exist some works based on Java EE for the construction of enterprise systems, such as [26], [27], [28], [29] or [30], just to mention a few, and specially for the creation of eLearning systems and applications, such as [31], [32] and [33].

In the [34] study, many benefits of the Java EE platform for the development of high quality web applications are enumerated, including usability, reliability efficiency and maintainability attributes among others. Other benefits of Java EE are:

- Lightweight implementations.
- Highly adopted, with a big support community.
- Stability. Many vendors collaborated in its definition and adopted it.

Java EE presents also some drawbacks:

- Steep learning curve.
- There is a lot of outdated and obsolete documentation about previous versions of the platform (J2EE).

III. RESEARCH METHODOLOGY

In order to select which of the preselected technologies is more suitable for the project, the following steps were applied as follows:

1. Study the current situation of each technology (the result of this step is shown in the previous section).
2. Evaluate some features of each technology.
3. Develop a prototype with each technology and evaluate results.
4. Integrate a formative tool with the developed prototype.

Next, the evaluated features of each technology and the design of the prototype are discussed in greater detail.

A. Evaluated features

The features of each technology to be evaluated in the second task are the following:

- Modularity. Ability to create modular applications.

- Security. Authentication, authorization and other security functionalities available.
- Inversion of Control. Work through an interface + dependency injection.
- Persistence. Ability to persist application data easily.
- Web applications. Ability to create web applications.
- Integration. Ability to integrate with other systems.
- Simplicity. Ability to create readable and understandable code, without extraneous and complex instructions or mechanisms.
- Ease of development.

B. *Prototype design*

The objective of the prototype is to implement a simple use case of the system using each framework in order to compare them. The selected use case is invoked by a formative assessment tool and receives the identifier of a user, returning the position on the tool leaderboard of that user. For the sake of simplicity, security is not handled at this stage.

The conceptual schema for the prototype is as follows:

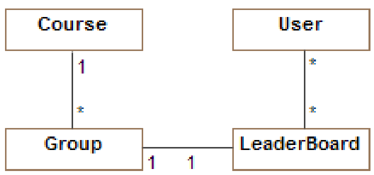

Figure 1. Prototype conceptual schema

And the sequence diagram for the particular use case selected for the prototype is the following:

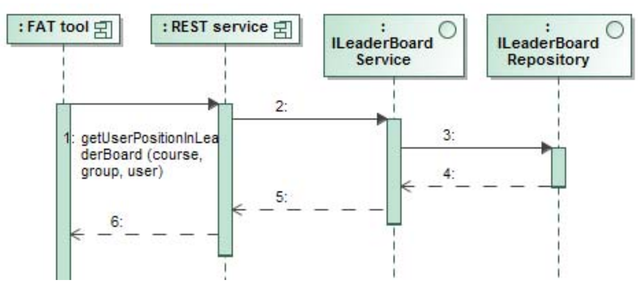

Figure 2. Prototype use case sequence diagram

For the evaluation of the prototype the following metrics were defined: (i) preparation time of the environment (in hours), (ii) training time (in hours), (iii) development time and (iv) performance.

The performance tests consisted in executing the use case 10.000 times measuring the total execution time, the maximum execution time, and the average. The test was repeated 8 times for each framework.

IV. EXPERIMENTATION AND VALIDATION

In this section, the results of the assessment methodology described in the previous section are presented.

A. *Features comparison*

The result of the evaluated features for each technology is presented below. Each feature is graded according to a qualitative scale that measures the adequacy of the technology for a particular feature (0 for poor support and 4 for excellent support). This grade is based on developer's knowledge and the information found during the study of each technology.

Table 1 presents the results for the OSGi technology. OSGi is a technology that enhances modularity and real encapsulation of modules. However, it is still a complex technology to work with because of the low maturity of the existing tools, and especially when developing web applications.

Table 1. OSGi technology features evalutation.

Requirement	Score (0-4)	Rationale
Modularity	4	- Enforces modularity by design.
Security	2	- No special support given. Uses Java APIs.
Inversion of Control	4	- Handled by the container. Little code is required, binding is done transparently.
Persistence	2	- Uses JavaEE APIs (JPA).
Web applications	2	- No special support given. Uses JavaEE APIs (servlets). Difficult to apply.
Integration	2	- Requires a specific server container. - Requires specific bundle files. Common jar files are not compatible directly.
Simplicity	3	- No extra code is needed to convert existing code into OSGi (only some metadata). - Project structure may vary from the common project structures (especially web bundles).
Ease of development	1	- Dependencies hell. Disperse help and outdated resources. Sparse adoption. Few good tools to ease bundle management.
TOTAL	**20**	

Spring is a well-known and wide spread technology. It is a good alternative to Java EE, especially to provide enterprise application features without a Java EE container (i.e., Spring can run in a simple servlet container like Tomcat). Table 2 presents the results for this technology.

Table 2. Spring technology features evalutation.

Requirement	Score (0-4)	Rationale
Modularity	3	- Does not enforce modularity by design. - A reference to the implementation jar is required.
Security	4	- Specific framework or use JavaEE APIs.
Inversion of Control	2	- Specific framework. Requires manual configuration.
Persistence	4	- Uses JavaEE APIs + Spring framework.
Web applications	4	- Uses JavaEE APIs + Spring frameworks like MVC.
Integration	3	- Does not require a specific server container, can run in a servlet container like Tomcat. - Requires more configuration and integrate external components and care of versions.
Simplicity	3	- Annotations greatly simplify its configuration. - A lot of XML configuration too.
Ease of development	2	- Good community and great adoption. - Outdated resources.
TOTAL	**25**	

Finally, Java EE is the specification for the construction and creation of enterprise Java applications. It includes several technologies and APIs that simplify the development tasks while provide many powerful features (Table 3).

Table 3. JavaEE technology features evalutation.

Requirement	Score (0-4)	Rationale
Modularity	3	- Does not enforce modularity by design but modularity can be achieved.
Security	4	- Specific API.
Inversion of Control	2	- Handled by the container. Not as good as in OSGi.
Persistence	4	- JPA
Web applications	4	- Servlets, JSP, JSF, JAX-RS, JAX-WS, etc.
Integration	3	- Requires a specific server container. More stable and mature containers than OSGi ones.
Simplicity	3	- Annotations greatly simplify its configuration.
Ease of development	3	- Lots of integrated IDE tools. Good community and great adoption. Outdated resources.
TOTAL	**26**	

If we compare the total score of each technology we achieve very similar results for Java EE and Spring (26 vs.25). OSGi gets a lower score of 20 mainly because of the low support for developers found, and the difficulty of developing web applications.

B. Prototype implementation

As explained in the approach section, the same prototype was implemented with each of the three presented technologies. Table 4 shows a time comparison for each implementation.

Table 4. Development time for each technology (in hours).

	OSGi	Spring	JavaEE
Preparation of the environment	1h	1h	1h
Training	22 h	12 h	8h
Implementation	20 h	10 h	8 h
TOTAL	43 h	23 h	17 h

Figure 3 shows the training and development time comparing to each other technology. As shown in the figure, the time required for the OSGi technology almost doubled the other two technologies. Java EE took little less time than Spring, although this can be partially due to the previous background of the developer.

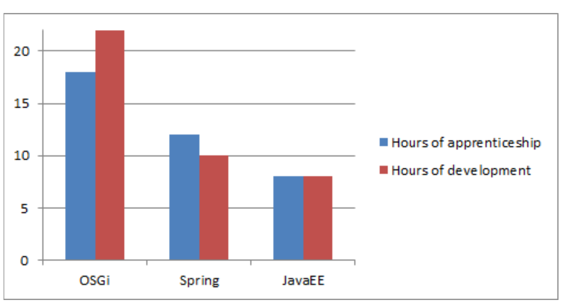

Figure 3. Development time for each technology (in hours).

In addition, a performance test of the prototype was conducted. The test consisted in 8 executions of a program that run the implemented use case 10.000 times. Figure 4 compares the aggregated average response time (AART) for all the executions in each technology. As can be seen in the picture, the Java EE solution had a better performance with respect to the other two alternatives.

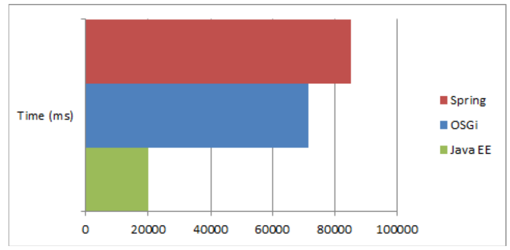

Figure 4. Comparison of the AART for each technology.

V. CONCLUSIONS

This paper presented the first steps into the development of our ICT-FLAG eLearning platform. The main goal of this research was to propose a methodology to select the most suitable development framework for our web-based e-Learning platform. Given that out-of-box solutions are discarded for customization and licensing concerns, several enterprise applications frameworks were evaluated, namely OSGi, Spring and Java EE.

OSGi is a very promising set of specifications for the modularization of applications. However, it is very difficult to implement in practice because of the lack of mature implementations and good documentation and support.

Spring is a well-known technology that appeared as an alternative to the first versions of the JavaEE specifications (known as J2EE) which were heavyweight and hard to develop. However, JavaEE has changed a lot since then for good and many best practices and concepts from Spring were applied to the specifications, improving JavaEE as a platform.

Nowadays, the main difference between Java EE and Spring is that the first one is a set of specifications, which are supported by multiple well-known vendors and there are multiple certified implementations, while Spring is just a set of libraries or framework with only a single implementation available.

Any of the compared software development frameworks and technologies can be used with more or less difficulties to meet the requirements of the project. In order to choose the most suitable one, a prototype was constructed with each of them, and a set of evaluation metrics were obtained and compared.

The data obtained from the study and evaluation suggests that Java EE is the more suitable technology than OSGi and Spring for the creation of the project, mainly because of the following reasons: (i) standard de facto technology; (ii) great community support; (iii) smaller development effort; and (iv) better performance.

We believe that this technology decision is also applicable to other projects that meet similar general non-functional requirements to our ICT-FLAG platform, such as high modularity, flexibility and performance.

Acknowledgments

This research was partly funded by the Spanish Government through the project TIN2013-45303-P "ICT-FLAG" (Enhancing ICT education through Formative assessment, Learning Analytics and Gamification).

References

[1] Whitelock, D. (2007). Computer Assisted Formative Assessment : Supporting Students to Become More Re ective Learners. In 8th International Conference on Computer Based Learning in Science, CBLIS (p. 492 - 503).

[2] Nicol, D., & Macfarlane-Dick, D. (2006). Formative Assessment and Self-Regulated Learning: A Model and Seven Principles of Good Feedback Practice. Studies in Higher Education, 31 (2), 199 - 218.

[3] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011, September). From game design elements to gamefulness: defining gamification. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9-15). ACM.

[4] Domínguez, A., Saenz-de-Navarrete, J., De-Marcos, L., Fernández-Sanz, L., Pagés, C., & Martínez-Herráiz, J. J. (2013). Gamifying learning experiences: Practical implications and outcomes. Computers & Education, 63, 380-392.

[-5] Amriani, A., Aji, A. F., Utomo, A. Y., & Junus, K. M. (2013, October). An empirical study of gamification impact on e-Learning environment. In Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference on (pp. 265-269). IEEE.

[5] Tiobe software. (2015). Tiobe Index for October 2015. Retrieved from http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html (accessed 14 October 2015).

[6] BuiltWith. (2015). Retrieved from http://trends.builtwith.com/framework (accessed 14 October 2015).

[7] Pierre Carbonnelle. (2015). Retrieved from http://pypl.github.io/PYPL.html (accessed 14 October 2015).

[8] Stack Exchange Inc. (2015). Retrieved from http://careers.stackoverflow.com/ (accessed 14 October 2015).

[9] Vogten, H., & Koper, R. (2014). Towards a new generation of Learning Management Systems.

[10] Shi, Y. Q., Zhou, J. W., Wang, P., & Tao, R. (2014, August). Research and Implementation of the Collaborative Learning Platform Based on Liferay. In Enterprise Systems Conference (ES), 2014 (pp. 331-335). IEEE.

[11] Blichar, J., & Podhradsky, P. (2012). Liferay as a collaboration and communication platform for m-learning subsystem. In Proceedings ELMAR-2012.

[12] Al-Jumeily, D., Murtada, A., Lunn, J., & Laws, A. (2011, December). Research 2.0: e-Learning Support Framework to Assist Research Degree Programmes. In Developments in E-systems Engineering (DeSE), 2011 (pp. 261-266). IEEE.

[13] Stalljohann, P., & Hackelöer, A. (2011, September). A rule-based group management tool-supporting organizational processes for collaborative learning activities in higher education. In Interactive Collaborative Learning (ICL), 2011 14th International Conference on (pp. 149-153). IEEE.

[14] Romero, C., Ventura, S., & García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. Computers & Education, 51(1), 368-384.

[15] Farmer, J., & Dolphin, I. (2005). Sakai: eLearning and more. EUNIS 2005-Leadership and Strategy in a Cyber-Infrastructure World.

[16] Liaw, S. S. (2008). Investigating students' perceived satisfaction, behavioral intention, and effectiveness of e-learning: A case study of the Blackboard system. Computers & Education, 51(2), 864-873.

[17] Godwin-Jones, R. (2014). Global reach and local practice: The promise of MOOCS. Language Learning & Technology, 18(3), 5–15. Retrieved from http://llt.msu.edu/issues/october2014/emerging.pdf

[18] Alliance, O. (2015). OSGi alliance. Retrieved from http://www.osgi.org/ (accessed 14 October 2015).

[19] Hall, R., Pauls, K., McCulloch, S., & Savage, D. (2011). OSGi in action: Creating modular applications in Java. Manning Publications Co.

[20] Santa, J., Úbeda, B., & Skarmeta, A. F. (2007, January). A multiplatform OSGi based architecture for developing road vehicle services. In Consumer Communications & Networking Conference.

[21] Lai, C. F., Ma, Y. W., Chang, S. Y., Chao, H. C., & Huang, Y. M. (2011). OSGi-based services architecture for cyber-physical home control systems. Computer Communications, 34(2), 184-191.

[22] Wu, C. L., Liao, C. F., & Fu, L. C. (2007). Service-oriented smart-home architecture based on OSGi and mobile-agent technology. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 37(2), 193-205.

[23] Colyer Adrian (2010, January 12). *Dm server projct moves to Eclipse.org.* Retrieved from http://spring.io/blog/2010/01/12/dm-server-project-moves-to-eclipse-org/ (accessed 14 October 2015).

[24] Arthur, J., & Azadegan, S. (2005, May). Spring Framework for rapid open source J2EE Web Application Development: A case study. In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on (pp. 90-95). IEEE.

[25] Oracle. (2013) Java Platform, Enterprise Edition (Java EE) Specification, v7. Retrieved from https://java.net/downloads/javaee-spec/JavaEE_Platform_Spec_PublicDraft.pdf

[26] Fan, W. (2010, June). A Java-EE based Product Design Management System. In Intelligent Computing and Cognitive Informatics (ICICCI), 2010 International Conference on (pp. 499-502). IEEE.

[27] Yongmin, Z., Pingjian, Z., Liangliang, H., Kai, W., & Qinqin, Z. (2012, January). Cartoon Resource Management System Development Based on the Lightweight Java EE Architecture. In Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on (pp. 796-799). IEEE.

[28] Huo, Y., Cui, Z., Liu, T., Zuo, X., & Wang, H. (2014, August). Design of Java EE-Based Remote Health Service System. In Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014 Sixth International Conference on (Vol. 1, pp. 339-342). IEEE.

[29] Caballé, S., Ortega, J.A., Camps, J.M., Barolli, L., Kulla, E., Spaho, E. (2014). A Presentation Framework to Simplify the Development of J2EE Application Thin Clients. In proceedings of the 8th International Conference on Complex, Intelligent and Software Intensive Systems, pp. 421-426.

[30] Liu, Y., Du, H., & Xu, Y. (2012, December). Design of Real-time Monitoring and Visualization Management System Based on RFID and Java EE. In Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on (pp. 569-572). IEEE.

[31] Brune, P., Leiser, M., & Janke, E. (2014, August). Towards an Easy-to-Use Web Application Server and Cloud PaaS for Web Development Education. In High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICESS), 2014 IEEE Intl Conf on (pp. 1113-1116). IEEE.

[32] Song, B., Zhang, Y., & Zhou, C. S. (2010, July). Implementation on network teaching system based on Java EE architecture. In Information Technology and Computer Science (ITCS), 2010 Second International Conference on (pp. 227-231). IEEE.

[33] Caballé, S., & Xhafa, F. (2010). CLPL: Providing software infrastructure for the systematic and effective construction of complex collaborative learning systems. Journal of Systems and Software, 83(11), 2083-2097.

[34] Prajapati, H. B., & Dabhi, V. K. (2009, March). High Quality Web-Application Development on Java EE Platform. In Advance Computing Conference, 2009. IACC 2009. IEEE International (pp. 1664-1669). IEEE.