DOI: 10.7763/IPEDR. 2012. V48. 38

A Web Academic Project Manager based on MVC Evolutionary Acquisition IRPM

Patrick Letouze [1 +], Mariwaldo G. Caetano [1], Fernando Chagas [1], Glaubos Climaco [1], and João Y. Ishihara [2]

[1] Federal University of Tocantins, Brazil – Department of Computer Science
[2] University of Brasília, Brazil – Department of Electrical Engineering

Abstract. Previously, we have developed an academic project management web system based on project management concepts, that we call SIGD. In this work, we present how we have applied the Model-View-Controller (MVC) pattern with Evolutionary Acquisition Interdisciplinary Research Project Management (EA-IRPM) for web services to implement a requirements analysis evolution web-based system for SIGD reusing its core, SIGD-RA. We have reused SIGD by mapping its elements like professor to analyst, and student to user, so each professor/analyst is responsible for a student/ (problem, suggestion, or requirement); also, each discipline represents a version of users' feedback, technological opportunities or evolving threats, for requirements analysis evolution. Additionally, one SIGD-RA may be connected to several SIGD, for instance in different Universities. Hence, in this paper we reported an academic project management web system evolution that has gone through refactoring, extensively reuse of source codes, which incorporates and manages user's feedback, technological opportunities and evolving threats to the system, which are also useful for systems maintenance. It is the implementation of a methodology that supports web services design, development, maintenance, and evolution. It is both a model and a method that takes advantage of the MVC architecture pattern for designing web system.

Keywords: Educational technology, Evolution, Evolutionary acquisition, IRPM, MVC, MVC EA-IRPM, Project management, Requirements analysis, Reusability.

1. Introduction

We have proposed a design architecture for web service in [1] that combines the Model-View-Controller (MVC) pattern [2]-[3] with Evolutionary Acquisition Interdisciplinary Research Project Management (IRPM) [4]. We have called this methodology MVC EA-IRPM, which intends to be an evolving web services development strategy triggered by users' feedback, and considerations of technological opportunities and evolving threats. In this work, we present its application to a real scalable system, an academic project management web system called SIGD.

We had developed SIGD 1.0 [5] to serve as a tool for improving results on a postgraduate programme. It has been based on project management concepts and its development had occurred in a classroom, where we had simulated a software house using role-playing [5] and a problem-based learning approach [6]-[7].

For its second version – SIGD 2.0, we had performed a code refactoring to facilitate improvements and the system's evolution [8]. We had employed agile development techniques and tools, together with the Evolutionary Acquisition Interdisciplinary Research Project Management methodology (EA-IRPM) [9] which is based on IRPM [10], the research strategy of NDS – Software Development Nucleus of the Federal University of Tocantins, where we are developing SIGD. Two other systems that make use of EA-IRPM are, respectively, a social network to provide free internet access for public schools' communities [11], and a medicine 2.0 architecture for managing transplantation patients [12].

In both versions, we had used MVC as the software architecture, mainly because it has been widely adopted to develop web services such as an integrated community webbased service platform [13], and a University project management model [14], which actually could be integrated to SIGD; and because it separates the business logic – Model, the user interface – View, and the user input – Controller. As a

+ Corresponding author. Tel.: + 55 63 32328027 ; fax: + 55 63 32328020.
E-mail address: patrick.letouze@gmail.com / letouze@uft.edu.br.

consequence, it provides a way to split functionalities to independent development, testing and maintenance. In other words, the Model represents application data and business rules that command data access and its modification, which also keeps business state and provides to the Controller the ability to access encapsulated functionalities. The View displays the system's state and the Controller sets the application behavior.

Now, we have implemented SIGD 2.1 using the concepts of MVC EA-IRPM. It integrates an independent database for users' feedback together with technological opportunities and evolving threats considerations that allows starting a new system's release for a dynamic evolution. Also, it allows the system to acquire new features or correct errors using users' feedback, that is, the Evolutionary Acquisition requirement analysis [15] becomes part of SIGD 2.1.

The authors organized this paper in accordance to the IMRAD structure: introduction, methods, results and discussion; which is adopted as part of the Uniform Requirements for Manuscripts Submitted to Biomedical Journals of the International Committee of Medical Journals Editors, 2008 update. The authors believe that adopting this structure would help search engines in international databases to store and to retrieve information within research papers in order to facilitate meta-analyses and systematic reviews.

2. Methods

In this section, we start by presenting the tools and techniques that we had used to develop SIGD's versions 1.0 and 2.0. Then we present our software development methodology.

2.1. Tools and Techniques

For SIGD versions 1.0 and 2.0 we had used MVC and Java as the object-oriented programming language, see Table 1. Moreover, we had used Javaserver Faces as the standard for building the user interface [16], PrimeFaces as the source Javaserver Faces components [17], Glassfish as the application server [18], and Netbeans as the integrated development environment (IDE) [19].

Then in SIGD 2.0 [8], see Table 1, differently from the first version, we had used Scrum as a process of iterative and incremental development for an agile software development [20], Hibernate as an object-relational mapping library for the Java language [21], Spring: security framework 3 as a framework for access control and authentication [22], Maven as a tool for management and automation of projects in Java [23], Prettyfaces as a filter-based servelets extension with support to JSF and to create URLs [24], and as the relational database management system, we had employed MySQL [25] and PostgreSQL [26].

As a starting point for refactoring SIGD 1.0, we have simplified the methodology presented in [27]. It is divided in preparation and search phases. The first phase consists of refactoring motives, to extract reusable components, to loose coupling between components, and to improve design. Then the second phase is: to select specifics parts of the project to do the refactoring; to propose new aspect-oriented program refactoring - the selection is done around the applicability and situation; to determine what refactoring activities can be applied; to set priorities for refactoring; to apply refactoring patterns, which implies in testing cases and realizing if it did not broke anything; and to analyze how the refactoring is affecting the software quality. However, SIGD 1.0 documentation was incomplete and its source code was poorly comprehensible, due that we had decided on a complete source code refactoring. Therefore, our motivation was to create a system that had a complete documentation, high comprehensibility of source code, both to prevent deterioration of its life cycle. Consequently, we had started with a design refactoring, which had considered aspects orientation for refactoring, then we had set priorities followed by case testing for finally analyzing quality improvements like simplification, code length and reusability [8].

Additionally, our second version – SIGD 2.0, had some improvements and corrections. Version 1.0 had proposed a system with unrestricted file uploads, initial module of internationalization, and the user could be enrolled only in one discipline as PhD thesis project. In contrast, the second version included a project area, chapters, restricted and validated upload, hierarchical changes to optimize searches, automatic generation of projects and chapters, internationalization, manual help, revision of heuristics, decreased response time and dynamic pages rendered via Primefaces' ajax. Now, a user can be enrolled in several disciplines avoiding the creation of multiple entries. For example, a research methodology discipline can also have a project.

Table 1: Comparing tools between versions 1.0 and 2.0

TOOLS	***SIGD 1.0***	***SIGD 2.0***
Software	MVC	MVC
Programming	Java	Java
Server	Javaserver	Javaserver
Application server	Glassfish	Glassfish
IDE	NetBeans	NetBeans
Development	Unified	Scrum
Object-relational	-	Hibernate
Application	-	Spring:
Build automation	-	Maven
URL rewrite filter	-	PrettyFace
Database	PostgreSQ	MySQL,

In relation to the frameworks used in version 2.0, Hibernate transforms classes in tables releasing the developer of manual work, while maintaining the system portable to any database. We had implemented it with MySQL to test the use of Hibernate, that is, it was enough a few parameters changes to make the system work with Postgre SQL, and that has confirmed the database abstraction [8]. Spring Security brings a complete solution for all kinds of security needs, and it automates security removing manual work which may be dangerous practice. About Maven, it is a simple basic project structure to code compilation and to run unit tests, and to build the jar file extension. For last, the URLs had become "friendly" with Prettyfaces.

SIGD version 2.1, which is the object of this paper, was developed with the same tools and techniques that we had employed in SIGD 2.0, except that instead of using MVC, we are using MVC EA-IRPM.

2.2. MVC Evolutionary Acquisition IRPM

The Interdisciplinary Research Project Management (IRPM) [10] is an approach for conducting interdisciplinary research of real problems using Project Management concepts [28] and problem-based learning [6], [7]. Evolutionary Acquisition (EA) is a system design methodology [15] and MVC is a software architecture pattern [2]-[3]. Their integration is called MVC EA-IRPM and to explain it, we start this subsection explaining IRPM followed by Evolutionary Acquisition resulting in EA-IRPM. In the sequence, we briefly review the MVC pattern to then present MVC EA-IRPM.

IRPM's schematic is present in Figure 1, and for a better understanding of it, let us review the Project Management phases [28]: (1) **Initiation**: to determine project goals, deliverables and process outputs, to document project constraints and assumptions, to define strategy, to identify performance criteria, to determine resource requirements, to define the budget and to produce a formal documentation; (2) **Planning**: to refine project, to create a work breakdown structure, to develop the resource management plan, to refine time and cost estimates, to establish project controls, to develop the project plan and to obtain the plan approval; (3) **Execution**: to commit resources, to implement resources, to manage progress, to communicate progress and to implement quality assurance procedures. (4) **Control**: to measure performance, to refine control limits, to take corrective action, to evaluate effectiveness of corrective action, to ensure plan compliance, to reassess control plans, to respond to risk event triggers and to monitor project activity; (5) **Closing**: to obtain acceptance of deliverables, to document lessons learned, to facilitate closure, to preserve product records and tools, and to release resources.

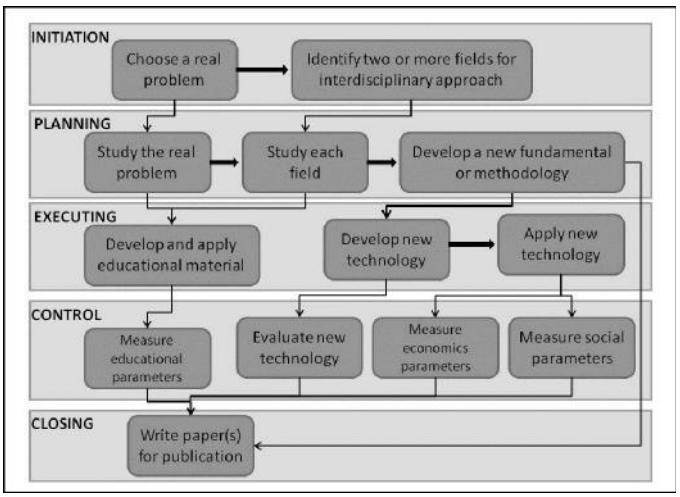

Fig. 1: IRPM – Interdisciplinary Research Project Management.

In IRPM, Initiation phase begins with choosing the real problem to solve and identifying at least two fields for an interdisciplinary approach. These fields are necessary to: document the real problem constraints and assumptions; define strategy; identify performance criteria; determine resource requirements; define budget; and produce formal documentation. Planning phase consists of refining project and analyzing the real problem through studying the chose fields. These studies may produce a new fundamental or methodology. Then in Execution phase, even if new concepts are not obtained, an educational material may be prepared and used in class for a problem-based learning approach, or else the new technology may be implemented and applied. Moreover, if in Planning phase controls were established then educational, technological, economics and social parameters may be available for measurement, allowing Control phase to be performed. Finally, after analyzing measurements, papers should be written as part of Closing phase.

Evolutionary Acquisition starts with the requirements analysis [15], see Figure 2. After defining the "general" requirements for the system and the "specific" requirements for the core, the concept of operations is elaborated. Then together with a requirements analysis of user feedback, technological opportunities and threats evaluation, the preliminary system architecture is developed. From the system architecture a core is produced. New definitions and developments with an operational test may result in a new version of the core. Then with experience and use, new requirements refinements and updates may be identified and used to develop a new core, or improve it. Additionally, Evolutionary Acquisition separates the core of the system into blocks. A particular block can have several releases. If the system is a software, then software engineering techniques may be applied. This paper proposes the use of MVC as the software architecture for the Interdisciplinary Research web service system.

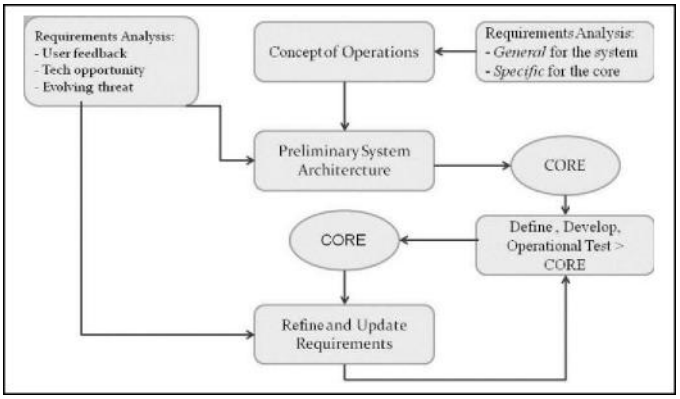

Fig. 2: The Evolutionary Acquisition model.

The incorporation of EA into IRPM is presented in Figure 3. It shows that it is inserted into phases Planning, Executing and Control, where RA means Requirements Analysis of: (1) general for the system and specific for the core; and (2) user feedback, technological opportunities and evolving threat. Hence, in Planning phase the attempt to develop a new fundamental or methodology consists of generating a

preliminary system architecture beginning with RA 1, and then elaborating the concept of operations, and when available, also considering RA 2. Executing phase consists of implementing the core from the preliminary system architecture followed by new definitions and developments of operational tests. Afterwards, the system is applied in a real life situation. Control phase is about refining and updating requirements, which implies in evaluating technology, measuring economic and social parameters, and verifying users' feedback, technological opportunities and evolving threats, that is, RA 2.

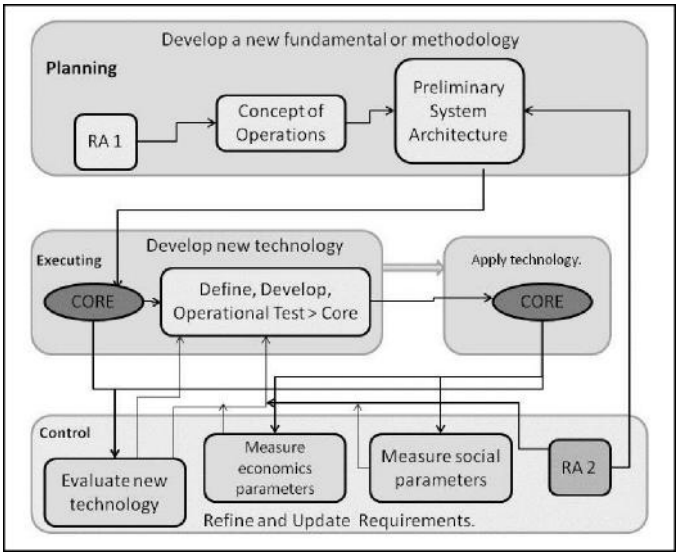

Fig. 3: The incorporation of the Evolutionary Acquisition model into the Interdisciplinary Research Project Management diagram.

The Model-View-Controller (MVC) pattern is a software architecture [2]-[3]. It intends to separate the business logic – Model, the user interface – View, and the user input – Controller. As a consequence, it provides a way to split functionalities to independent development, testing and maintenance. Basically, the Model represents the application data and the business rules that command data access and its modification. It also keeps business state and provides to the controller the ability to access encapsulated functionalities. The View displays the system's state and the Controller sets the application behavior.

The idea of incorporating MVC into Evolutionary Acquisition is presented in Figure 4. The core architecture of the web service is a modified MVC pattern that is connected to the Requirement Analysis – RA2 of Figure 3, through the user feedback, which should be an independent database system. The letter links means: *A* – to query the model state; *B* – to notify view of change in model state; *C* – state view; *D* – user actions/commands; *E* – invoke methods in the models public APIs; *F* – output to user; *G* – input from user; *H* – to report problem/suggestion/requirement (psr). Connection between web services' core and Requirement Analysis occurs in the following way: (1) user identifies a system's psr; (2) then user access the view to report psr, for instance, by pressing a specific button available in the user interface; (3) View queries Controller state about psr; (4) Controller notifies View of change to psr state; (5) View displays psr state to user; (6) user reports psr through View; (7) View transmits user's report to Controller; (8) Controller accesses users' feedback database system to report psr; (9) refinements and update requirements are defined using users' feedback, technological opportunities and evolving threats considerations; (10) a decision to start a new release may be taken. Hence, MVC is also incorporated to the Evolutionary Acquisition IRPM, which is presented in Figure 5. It is worth noticing that a new release may imply in a change in the system's architecture or a new block release.

The decision of releasing a new version of the core or a block is not automatic, and considerations of technological opportunities and evolving threats are still independent parts, that is, they are not necessarily part of the users' feedback database system, though they might be and they actually are in SIGD 2.1. Moreover, they are not automatic too. Consequently, error and fault detection mechanisms are important features to be considered for future developments, some works that may be applied are in [29]-[31].

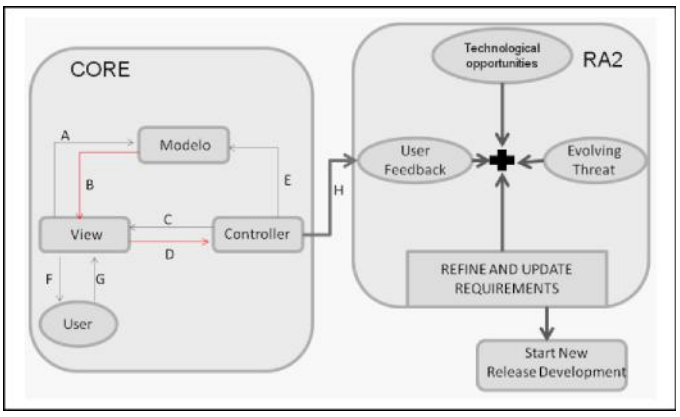

Fig. 4: The Model-View-Controller Evolutionary Acquisition.

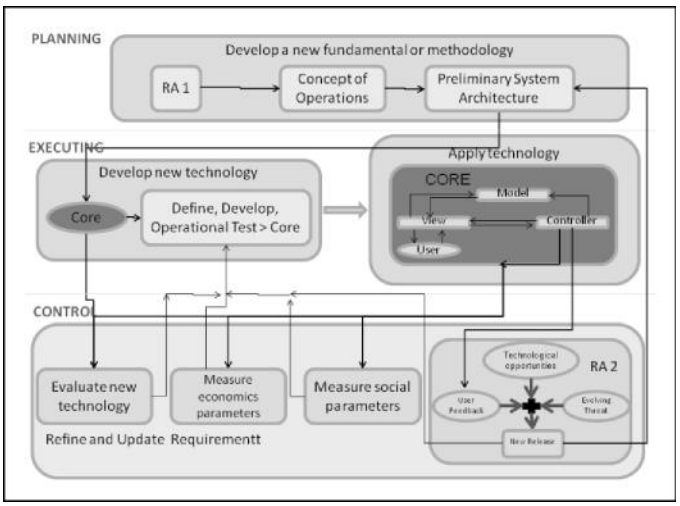

Fig. 5: The Model-View-Controller Evolutionary Acquisition Interdisciplinary Research Project Management diagram.

Hence, MVC EA-IRPM presents evolution as an important factor in interdisciplinary web services systems. Actually, according to Breivold *et al.* in [32] "the ever-changing world makes evolvability a strong quality requirement for the majority of software architectures", that is, MVC EA-IRPM intends to increase productivity and to facilitate software evolution.

3. Results

The new version of SIGD, version 2.1, has two parts (http://comp.uft.edu.br/rgm). The first part is the academic project management system that we call from now on *SIGD_i*, where *i* is an index number denoting different systems that may or may not be located geographically apart. The second part that we denote from now on *SIGD-RA*, which is a modification of SIGD that implements a database for the requirements analysis such as RA 2 in Figure 4.

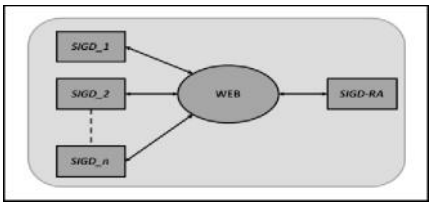

Fig. 6: Web structure for the SIGD system.

SIGD-RA is a web-based system that communicates with every registered *SIGD_i*. That means that every institution may have its own SIGD, and for requirements evolution, it communicates with only one *SIGD-RA*. The communication is as described previously in Section II – B. For the user, it seems that he or she is using its own system, because communication is incorporated in the system's interface, that is, the MVC EAIRPM model has only one View. Therefore, scalability is assured for the SIGD system, as presented in Figure 6.

SIGD_i's interface is as presented in Figure 7. In its menu there exits the User Feedback option which gives access to the *SIGD-RA* system – Figure 8. When accessing that option, the user may report a system's problem, suggestion or requirement as in Figure 9. Additionally, it may be really a user's feedback, or a technological opportunity consideration, or an evolving threat identification, see Figure 10. From the system's point of view in *SIGD-RA*, the requirements analysis is a project such as an academic project in a regular SIGD, that is, we have completely reused SIGD's core to develop a new application or web service. Besides, user's reports are managed through the system – *SIGD-RA*, Figure 11.

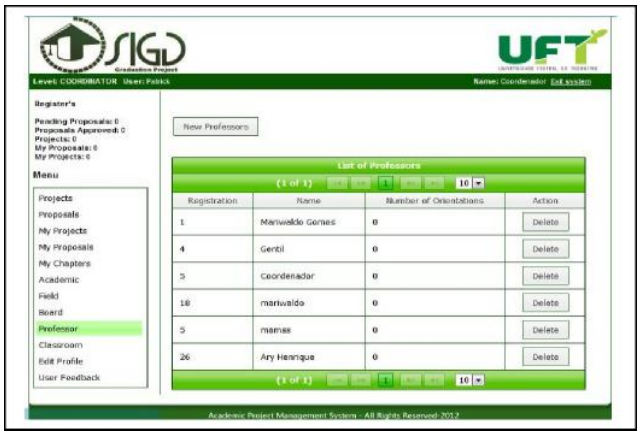

Fig. 7: *SIGD_i* system's interface.

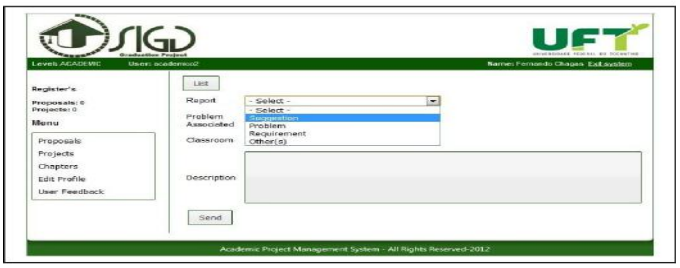

Fig. 8: User's feedback interface; and Type of report: problem, suggestion or requirement.

Fig. 9: Type of requirement: user's feedback, technological opportunity or evolving threat.

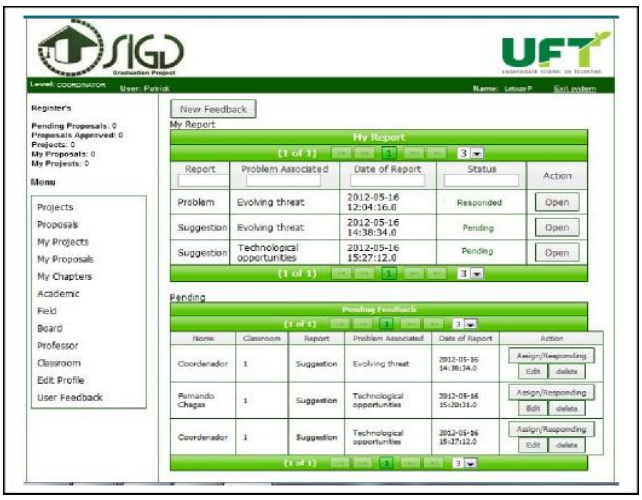

Fig. 10: Web structure for the SIGD system.

4. Discussion

Requirements evolution plays an important role in Web Systems Evolution. Besides, software architecture evolution is relevant for extending software lifecycle [32]. The MVC Evolutionary Acquisition IRPM combines these to features, that is, it is a software architecture that incorporates requirements analysis in its model. More than that, it aggregates project management concepts with interdisciplinary research.

In a systematic review of software architecture evolution, Breivold *et al*. in [32] concluded that "it is necessary to establish a theoretical foundation for software evolution research" and "to combine appropriate techniques to address the multifaceted perspectives of software evolvability". This work is the implementation of a methodology that combines well-established techniques for web systems evolution.

MVC EA-IRPM supports web services design, development, maintenance, and evolution. It is both a model and a method that takes advantage of the MVC architecture pattern for designing the web system, of the Evolutionary Acquisition for developing systems continuously using requirements analysis together with systems testing, and of project management concepts in an interdisciplinary research context through IRPM.

Hence, in this paper we reported an academic project management web system evolution that has gone through refactoring; that has extensively reused source codes; that incorporates and manages user's feedback, technological opportunities and evolving threats to the system, which are also useful for systems maintenance. For future works, we intend to adjust *SIGD-RA* for academic purposes,which means a web service to support not only academic projects evolution, but also the evolution of postgraduate programs, which could provide important information to a Ministry of Education.

5. References

[1] P. Letouze, M. G. Caetano, J. Y. Ishihara, D. Prata, and G. Brito. "Applying MVC to evolutionary acquisition IRPM," *International Proceedings of Computer Science and Information Technology*. [in press]

[2] T. Reenskaug, "Thing-Model-View-Editor – an example from a planning system," Technical note, Xerox PARC, May 1979. A scanned version on http://heim.ifi.uio.no/~trygver/mvc/index.html.

[3] T. Reenskaug, "Models-Views-Controller," Technical note, Xerox PARC, December 1979. A scanned version on http://heim.ifi.uio.no/~trygver/mvc/index.html.

[4] P. Letouze, "Incorporating Evolutionary Acquisition into Interdisciplinary Research Project Management," *IPEDR*, 2012, vol. 30: 338-342.

[5] P. Letouze, R. A. Ronzani and A. H. M. Oliveira, "An Academic Project Management Web System Developed through a Software House Simulation in a Classroom," *IPEDR*, 2011, vol. 14: 587-592.

[6] J. R. Savery. "Overview of Problem-Based Learning: Definitions and Distinctions," IJPBL. 2006, **1** (1): 9-20.

[7] O. Pierrakos, A. Zilberberg and R. Anderson. "Understanding Undergraduate Research Experiences through the Lens of Problembased Learning: Implications for Curriculum Translation," IJPBL. 2010, 4 (2): 35-62.

[8] F. Chagas, G. Climaco, M. G. Caetano, A. H. M. de Oliveira and P. Letouze, "Reporting a code refactoring and evolution of na academic project management web system," *ICAICTSEE 2012 – 13th International Conference on Application of Information and Communicatio Technology and Statistics in Economy and Education*, Sofia, Bulgaria, Ocober 5-6th, 2012.[submitted]

[9] P. Letouze, "Incorporating Evolutionary Acquisition into Interdisciplinary Research Project Management," IPEDR, 2012, vol. 30: 338-342.

[10] P. Letouze, Interdisciplinary Research Project Management. *IPEDR*, 2011, vol. 14: 338-342.

[11] G. V. Barbosa, H. D. Soares, G. L. R. de Brito, Sergio M. S. da Cruz, and P. Letouze, "A Social Network to Provide Free Internet Access for Public Schools' Communities," IPEDR, 2012, vol. 30: 248-253.

[12] P. Letouze, S. A. Naimayer, F. S. Nunes, G. V. Barbosa, A. H. M. de Oliveira, J. Y. Ishihara, and A. Burls, "A Medicine 2.0 Architecture: managing transplantation patients," [to be submitted].

[13] W. Xiaoyan, and W. Fang, "Comprehensive Service Platform of Integrated Community Based on the Web," Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, July 27-30, 2011, pp. 1208-1211.

[14] L. Zhang, F. Ma, C. Zhong, L. Zhang, and Y. Wang. "A MVCD Model and Its Application in University Project Management," Proceed. 6th World Congress on Intelligent Control and Automation, 2006, pp. 7008- 7012.

[15] "Systems Engineering Fundamentals," Supplement 2-B "Evolutionary Acquisition Considerations," The Defense Acquisition University Press, Fort Belvoir, Virginia, Jan 2001.

[16] James Holmes , Chris Schalk, JavaServer Faces: The Complete Reference, McGraw-Hill, Inc., New York, NY, 2006, page 3

[17] http://www.primefaces.org/

[18] http://glassfish.java.net/

[19] http://netbeans.org/

[20] Schwaber, Ken. 2004. Agile Project Management with Scrum. Redmond, WA. Microsoft Press. (ISBN 0-7356-1993-X)

[21] Will Iverson, Hibernate: A J2EETM Developer's Guide, 2004, Chapter 1. Overview, Whats is hibernate?

[22] Peter Mularien, Spring Security 3, Packt Publishing, 2010, page 18

[23] http://maven.apache.org/

[24] http://ocpsoft.org/prettyfaces/

[25] http://www.mysql.com/

[26] Official Documentation. *PostgreSQL 8.4 – Volume III. Server Programming*. Fultus Corporation, 2009.

[27] S. A. M. Rizvi, Z. Khanam, "A methodology for refactoring legacy code," *Electronics Computer Technology (ICECT), 2011 3rd International Conference on* , vol.6, no., pp.198-200.

[28] K. Heldman. PMP Project Management Professional Exam Study Guide. Sybex, 2009.

[29] H. Xu, W. Chen, and H. Qian. "Research on error feedback mechanism of information system," 2nd International Conf. on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011, pp. 5112-5115.

[30] K.-L. Cheng, C.-P. Chang, and C.-P. Chu. "Software fault detection using program patterns," IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), 15-17 July, 2011, pp. 278- 281.

[31] H. Okamura, Y. Etani, and T. Dohi. "Quantifying the effectiveness of testing efforts on software fault detection with a logit software reliability growth model," IWSN-MENSURA, pp. 62-68.

[32] H. P. Breivold, I. Crnkovik, and M. Larsson. "A systematic review of software architecture evolution research," Information and Software Technology, 2012, vol. 54: 16-40.