The Impact of Open Source Software on an Educational Business Model

V. L. Plantamura¹, A. Marengo² and A. Pagano³

¹ University of Bari/Computer Science Department, Bari, Italy
² University of Bari/Computer Science Department, Bari, Italy
³ University of Bari/Computer Science Department, Bari, Italy

Abstract— The Free and OSS movement has had phenomenal impact on the industry’s evolution, in fact most of the companies today, makes extensive use of Open Source software and technologies. The research communities, are engaged in the study of the open software-related development in order to highlight the advantages and disadvantages in terms of technology, reuse and economic impact. Nowadays there are no methodological best practices that set standards to determine the cost/benefit analysis for this approach. After a deep analysis of the impact that the implementation of an educational Open Source software has in workplaces, it’s possible to develop a characterization model harmonizing the variables and critical factors used in business and market contexts, and assess its effectiveness empirically. As stated in literature, the OSS phenomenon has promoted research, and everyone knows that educational field is a permanent research laboratory and when it is supported by computers (CSCL), the challenge is to find the right mix between didactic methodology and technological resources. Open Source approach stimulates viral innovation. Despite the identification of factors and variables that contribute to weigh this choice, it is not possible to define a model of characterization that can help to address a sensitive junction yet (even because there’s not enough empirical data). It need to define a model that could harmonize the largest number of variables, in order to give a valid aid to corporate management. The research team involved in this field, is experimenting on empirical basis, a model of characterization for the selection of educational software tools belonging to the different needs. The case of study that could be suitable to be showed is the Osel project (Open Source e-learning) experience, in which this model is applied and that brought successful results in terms of educational effectiveness and power of software.

Index Terms— e-learning, Education, Open Source, Software.

I. INTRODUCTION

Nowadays the tools to deliver e-Learning courses can facilitate distance-learning activities. In the open source field, although the existing LMS (Learning Management System) have all the features required to deliver on-line courses (registration of students, management of training contents, evaluation of the knowledge, etc.), they don’t have a system of intelligent tutoring that can help both the teacher that the student in the development of dynamic courses, starting from a set of learning goals.

The main topic is to analyze the features, the standards, and the structure to be used for the implementation of Web Intelligent Agent (LIS – Learning Insight System) that can interact with the existing open source LMS, expanding their traditional features with the innovation of intelligent tutoring.

The aim of this research project is to develop an innovative approach in the learning field and evaluate the impact on the learners and teachers.

This project started from the necessity to create a flexible and integrated system, based on Open Source software, in which an Intelligent Web Agent will be integrated to manage the Learning Objects repository and LCMS. This approach taken provides high scalability and versatility for the system and streamlines the upgrading process. It aims at meeting some changeable requirement in the field of distance learning (yearly or even monthly innovation). The modular structure and flexibility provided by the portal developed makes this project adaptable to any SCORM compatible LCMS. Specifically, the OSEL’s repository is the integration between Learning repository software LeMill, with OSEL Taxonomy for the classification of Learning Objects (OSEL stand for Open Source E-Learning and it’s a research project of University of Bari).

Integration between software and learning structure will improve portal features and teaching/learning process. The flexible and modular structure of our server and our Learning Environment based on Zope (Zope is a free and open-source, object-oriented web application server), Plone (Plone is a free and open source content management system built on top of the Zope application server), and Fle (Fle is a Web-based learning environment or virtual learning environment) with integration of LeMill (Web community for finding, authoring and sharing learning resources) repository, provide an environment suitable for Intelligent Web Agent applications. Artificial Intelligence could rewrite the rules of learning online.

An Open Source approach is not necessarily a no-cost solution, so what concerns might a company have that would adopt/develop an Open Source framework like this, when considering its business model? What’s factors should management analyze before choosing an Open Source solution?

This chapter describes the technological structure of OSEL LIS as the first step of our research framework. Then the research team tries to analyze the open source business model that guided crucial choice in adoption/development of the same framework from the point of view of a research-oriented company.

II. THE PROJECT

There has been much debate recently about the use and benefits of virtual learning platforms, however they generally fail to actively support users on an individual
basis to actually help them learn at a pace that is appropriate to the learner and that can identify knowledge or skill gaps and address them dynamically.

The aim of the “OSEL 2.0” project is to develop such an approach and evaluate the impact on the speed and success of learning given specific learning outcomes (skills/knowledge gained by the student and new methodologies applied by the teacher).

The research framework proposed will include not only a methodological and theoretical research on an “intelligent” LCMS and LO repository, but also its technological project and its integration on an extended platform.

The research framework started on January 2008 with the definition and implementation of the technological structure of the server, based on Open Source software as Zope, Plone and Fle. It will move on this path having as target the implementation of the, so called OSEL LIS - Learning Insight System that will be completely web-based and its innovative distinguishing marks will be:

Automatic or assisted building of learning paths, starting from the learning goals;

Automatic customization of each course of study based on the knowledge of the student and his learning likes (preferences, ways to learn...);

Monitoring and automatic evaluation of student’s knowledge related to acquired information and cognitive skills;

Content management through the use of ontologies, compliant to most important international standards on Knowledge modeling (Devedzic, 2006).

The modular structure and flexibility provided by LIS will make this system adaptable to any kind of educational and/or academic situation, allowing also the development in step with innovative and specific Web technologies.

Updating ability and flexibility towards the necessity of implementing innovation into technology as well as educational and meta-cognitive methodologies were the basic criteria for choosing software, which is used to develop the framework.

Furthermore, this project will develop such an approach using semantic web technology and reasoning support and evaluate the impact of this new approach against rival techniques/systems identified in the literature. A usability study is set up to evaluate this.

III. FRAMEWORK STRUCTURE

One of the most important topics in Open Source development, today, is the integration of already developed components. The critical topic is to choose the software (or pieces of software) to integrate for the final product.

- Linux Server (Gentoo with xen virtualization).
- Client (any OS) + Browser.
- Zope
- Plone.
- Fle.
- LeMill.
- LIS

The software structure (Figure 1) is modular / onion-skin, in fact the core is the programming language (Python), then we have Zope, which works as a strong and stable web server and then, at the upper level, we can find Plone and Fle which are the effective user interface. LeMill is the Learning Object repository and LIS is the Intelligent Agent that structure data in ontology ways and build personalized learning paths.

IV. AGENT SYSTEM IN LEARNING ENVIRONMENT

According to Dolonen, Chen, and Mørch and their DoCTA project, implementing their thought experiment, we can develop an intelligent software agent that could work in Learning Environment FLE (Dolonen, Chen, & Mørch, 2003).

In the process of collaborative knowledge building, it is usually difficult for students to be aware of others’ activities and for instructors to overview the process and to regulate the collaboration In order to facilitate collaborative knowledge building, intelligent agents were developed to support the awareness and regulate the collaboration (Palade, Howlett, & Jain, 2003)

Instead of letting the agent contact the students directly, which can be inappropriate and annoying, the Intelligent Web Agent was designed and developed to assist the instructor in giving feedback to students. This agent would instead present statistical information and advice to the instructors to inform them about the collaboration process. Then an instructor could, if judged appropriate, forward the feedback and decide to engage in a dialog with the student. In this way, the instructor retains a role in the success of collaborative learning. However, to accomplish this role the instructor will need specific tools for monitoring interactions that are distributed in time and space. The design of these tools is very important for CSCL research (Dillenbourg, 1999).

The Intelligent Web Agent is such a tool for enhancing the facilitator’s ability to monitor and regulate the process of collaboration and knowledge building. (Dolonen et al., 2003)

V. THE ROLE OF LEARNING REPOSITORY

New publishing methods require new approaches to traditional copyright laws: all resources are freely usable by anyone in any context (we can imagine Youtube videos or Slideshare slides). All the content in LeMill platform is released under Creative Commons Attribution-ShareAlike 2.5 (Creative Commons, 2009).

The emerging questions related to Learning Repositories are: What are the success factors and obstacles for collaborative authoring of learning resources by communities of practice? What are the emerging patterns in social software that support collaborative authoring of learning resources?

The repository implemented has the function to catalogue and to research the Learning Objects (LO) at the user’s disposal, i.e., a sort of warehouse of LO similar to a database in which LO are registered and classified. The SCORM standard and the flexibility of Python language makes the didactic materials re-usable in different situations and in different platforms.

VI. FUTURE DEVELOPMENT AND FUTURE LEARNING ENVIRONMENT INTEGRATION WITH LE MILL REPOSITORY.

Integration between FLE and LeMill is the next step for improving platform features. Using a web agent similar to DoCTA with advanced AI algorithm, users will easily
access to proper LeMill contents, directly from FLE environment. We could imagine a system that provides personalized set of learning contents for each user, depending on his own skill level.

VII. WEB INTELLIGENT AGENT (LIS) AS SUPPORT FOR E-LEARNING COURSES

LIS (Learning Insight System) will be based on formal models of domain able to represent knowledge and didactic experience, structured through the use of ontologies and encoded with the standard OWL (Web Ontology Language) by W3C (World Wide Web Consortium).

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies and is endorsed by the World Wide Web Consortium. OWL is considered one of the fundamental technologies underpinning the Semantic Web, and has attracted both academic and commercial interest.

LIS will perform a logical inference on ontologies, using a typical behavior of intelligent systems based on knowledge (Knowledge Based System).

It will use a repository of Standard Learning Object compatible with SCORM. The Sharable Content Object Reference Model (SCORM) is a collection of standards and specifications for web-based e-learning. It defines communications between client side content and a host system called the run-time environment (commonly a function of a learning management system). SCORM also defines how content may be packaged into a transferable file.

This approach will facilitate the interoperability of training materials with all the systems that support this standard.

The agent will be developed with a structure based on services (SOA) by displaying a public interface through the use of Web Services. The services will be described through WSDL standard and the messages will be coded according SOAP standard.

Using standards for the services exposure allows interoperability of LIS with all the e-Learning platforms; to use all the features of intelligent agent, it will be enough for each e-Learning platform to implement an additional module (plug-in) that will provide an interface for the user and it will communicate with the staff calling services through Web Service. Additional modules will be developed for open source e-Learning platforms such as Moodle and FLE.

It is easy to deduce that LIS, designed as knowledge-based system, uses structured ontologies following OWL standard and exporting services through WSDL standard, can be considered a system ready for Semantic Web (Web 3.0).

VIII. FUNCTIONAL REQUIREMENTS

Below are described the functional requirements of Intelligent Web Agent as support to e-Learning platforms through the UML diagram of Use Cases.

A. Use Case: Add Knowledge through Ontologies – Upload Learning Object

The Learning Object writer reaches the user interface made available by LIS, and following a set of guided steps, proceeds to build the ontologies (with selection of individuals) through OWL language and the following upload of Learning Objects SCORM linked (Figure 2).

B. Use case: Adaptive access to learning object repository

In the creation stage of training course in the LMS (e.g. Moodle or FLE), LIS intervenes through its plug-in installed in e-Learning platform and drawing from the repository, offers to the Teacher a choice of appropriate Learning Object, to use in case together with the other Learning Objects chosen (autonomously) by the Teacher.

C. Use case: Making courses starting from educational targets (case 1)

In the creation stage of training course in the LMS (e.g., Moodle), after the Teacher sets the didactic goals, LIS intervenes through its plug-in (with questionnaires and forms) and, drawing from its own repository of ontologies and Learning Object, suggests to the Teacher all the Learning Objects useful to the creation of the course.

D. Use case: Intelligent suggestion of extra educational contents

During the attendance of on-line course in the LMS, the student may have the need to deepen his knowledge through additional contents.

LIS in this case provides a virtual tutor able to offer additional training materials (Learning Object, web searches, etc.) consistent with didactic contents of (related to) the course.

E. Use case: Making courses starting from educational targets (case 2)

This scenario involves a self-taught student who has learning goals to achieve, but who does not know how to structure an appropriate learning path.

In this context, LIS, through its plug-in in the LMS, will perform inference on ontologies put in its repository and will propose to the student an appropriate learning path in order to achieve his training goals.

F. Use case: System monitoring and supervision

This scenario involves system administrators, systems analysts, and developers who are responsible for maintaining and monitoring the system.
IX. LIS INTELLIGENT WEB AGENT STRUCTURE

The following proposed deploy diagram (Figure 3) displays the structure of intelligent Web Agent and how it communicates with the client and with the LMS:

The deploy diagram shows that the intelligent agent, based on J2EE platform, will be installed on a dedicated server.

Learning Object, ontologies, and all supporting data that are essential for the right functioning of LIS will reside in internal specific repositories and database and represent the knowledge on which to base logical inference. Furthermore, the agent will communicate with the repository of SCORM compliant learning object (Lemill) to obtain additional materials.

The LMS, which are installed on separate servers, can use the features of an intelligent agent communicating through web services interfaces.

LIS features will be presented in a transparent way to the final user through a plug-in installed in the LMS used by the user. In fact, the final user (teacher and student) who wants use LIS features will just need a web browser to join his open source LMS; it will be the plug-in to communicate through web services with LIS. Finally, the creator of Learning Object will direct join LIS through HTML interface specially developed that will help him to enter Learning Objects and knowledge.

X. ONTOLOGY AND TAXONOMY

The necessity to find a flexible taxonomy for the LO had inevitably led to issues related to ontology.

The introduction of ontologies in the computer science world gives a valid tool to the learning process. Above all, in the A.I. context the use of them is actually increasing for the significant role they have in information systems, in the semantic Web, and in the systems based on knowledge, as for instance a neural net. The recent attention the A.I. community is paying to ontologies focuses on the theories about content more than those about mechanism. Chandrasekaran, Josephson, and Benjamins (1999) suggest that, although mechanisms are important for the functioning of intelligent machines, they are useless without a good theory of content on which mechanisms must rule. Furthermore once a good theory of content is available, different mechanisms can be used to implement efficient systems ruling on the same content.

Thus the ontologies become theories of content as they contribute to identify specific sets of objects and relationships that exist in a specific domain of knowledge.

XI. OSEL TAXONOMY

Starting from the awareness of the lack of a universally recognized taxonomic classification, this research has been oriented to study not only the structural characteristics of any single LO but also their interoperability with the users participating, or not participating, in a group activity.

The OSEL Taxonomy is based on the two most significant taxonomies, known all over the world. The first is Wiley’s taxonomy, called “Preliminary Taxonomy of Learning Object Types” (Wiley, 2000). The second taxonomy is based on the “Educational Taxonomy for Learning Objects” (Redeker, 2003). It focuses above all on the didactics aspects related to the LO.

The aim of the OSEL Taxonomy is to classify the LO that can be used within a LCMS platform and, thus, just those LOs that can be re-used.

The OSEL Taxonomy classifies the LO both through an ontological definition related to their domain of competence and through the relationship that could eventually exist among them and the learners without delegating subjective opinions to the author. The extremely accurate construction of a glossary, based upon the re-usability concept as ontological requirement of the LO, makes the OSEL Taxonomy particularly efficient for the classification of the LOs that are used within the LCMS platforms (Convertini, Albanese, Marengo, Marengo, & Scalera, 2006).

XII. AI AND NEW OSEL APPROACH

The next step of development is to improve the interoperability between learning environment and learning object repository, implementing OSEL Ontology with new networked taxonomy. We can imagine an Artificial Intelligence web agent guided by Description Logics and OWL based algorithm, which could adapt learning path to every student.

An important topic for this research is represented by interoperability between platforms. This is reached with full standardization and full modular structure, starting from software, ending to logical approach. We can imagine three level of Knowledge Model:

- Learning Objects;
- Metadata;
- Ontology.

This approach gives to the web agent more detailed information about didactic material, but even deeper information about the relationship between Learning Objects, metadata, and student’s skill level.

With this kind of approach the Intelligent Agent could select Learning Object from repository to build a personalized learning path.

The development of AI module for our platform is just began, we think that it could be developed as Zope module (written in Python) to include it easy in our learning environment and to maintain high versatility, modularity and suitable for frequent updates, bugfix, and codes improvement.
XIII. OPEN SOURCE SOFTWARE IMPACT ON EDUCATIONAL BUSINESS MODELS

Open Source approach is not a no-cost solution, so what about a company that would adopt/develop an Open Source framework like presented one, in its business model? What’s the analysis that the management should do before choosing an Open Source solution? 

A. The Open Source Business Model in Research Oriented Firms

The Free and Open Source Software movement has had phenomenal impact on the industry’s evolution, in fact most of the companies today make extensive use of Open Source software and technologies. The research communities, academic and professional, are engaged in the study of the open software-related development in order to highlight the advantages and disadvantages in terms of technology, reuse, and economic impact.

Below are described and analyzed the experience in the decisional process during the research & development of the Open Source e-learning framework with LIS.

Educational environments based on technology, could be considered as a real information systems. An organization that would adopt a Learning Environment, needs to face the knowledge management problem.

As a result of recent research, there is now a growing understanding of the drivers of information systems development and performance, and methods are evolving to improve the delivery of appropriate learning environments that return real knowledge benefits to the learners (Remenyi, White, & Shervood Smith, 1997).

Nowadays there are only a few methodological best practices that set standards to determine the cost / benefit analysis for this approach. Today’s research has an ambitious aim: harmonizing the variables and factors critical in implementing an Open Source model and placed in a model built on the experience of previous business and the environment inside and outside the company.

Similarly to the SWOT (analysis of Strengths, Weakness, Opportunities, Threats) approach, after analysis of the impact that the implementation of an Open Source software has in a company, it is possible to develop a characterization model used in business and market contexts and assess its effectiveness empirically.

The Open Source software component used to develop the framework is adopted for both ideological and purely pragmatic reasons (Ven, Verelst, & Mannaert, 2008), but it is necessary that the strategic management focus the attention on some critical topics that may influence this choice: a decision not weighted, can cause damage to the project or make the research team miss opportunities in terms of profitable results and development. The variables involved are numerous and not always immediately visible at first analysis.

B. An Open Source Research Project Work in Free Market

The framework developed is suitable for many environments and probably its suitable place is free market. The question is: what about a company that would adopt (or even develop) this Open Source Framework?

The Company needs to consider an important factor that makes it sensitive in evaluating investment: the cost-benefit analysis. Often, the OS software seems free, and this could bring the management to wrong decision in economy and market developing fields.

The OS software is, in most cases, distributed for free, but it is wrong to think that the company don’t have to pay for its adoption, but it is useful to point out all the costs necessary to use as a real economic asset for enterprise business. Not all OSS is free, so OSS might not be less expensive than proprietary software. To estimate the costs involved in introducing OSS, an organization can calculate the TCO (total cost of ownership) (Ven et al., 2008)

The free availability of source code can create advantages or disadvantages in the technical environment and in the economical environment. Depending on the context, activity, and the type of target market, the educational organization can be classified into three categories:

Open code indifferent: In this scenario, the source code availability is neither an advantage nor a disadvantage for the organization. OSS serves as a black box and its advantages or disadvantages are comparable to proprietary packaged software.

Open code scholar: In this scenario, the organization considers the source code availability to be an advantage, but doesn’t use it to study or customize the program. Some organizations choose OSS because they feel that the program is less likely to contain hidden features or bugs, and that in case of bug discovery, it will be fixed quickly. With this kind of approach the use of OSS implies a learning process, in which the organization gains experience and skills (and this could represent a profitable investment)

Open code developer: In this scenario, OSS serves as a white box (Weinstock & Hissam, 2005). Organizations use the source code to study the software’s inner workings or to adapt the software to their own (or their clients) needs. This is primarily interesting for software houses developing OSS-based applications.

This third kind of approach introduces the theme of OSS reuse (Ebert, 2008). Adopting Open source development practices can make an organization pay less attention to strategic planning, detailed requirements elicitation, testing, and organized support (Spinellis & Syzoperski, 2004).

In the framework case described, the choice to use Open Source Software is obvious. The cost that a company or a research group may pay is represented by TCO (Total Cost of Ownership). The reuse of already developed content is cheaper than the development from scratch. In e-learning field the Open Source Software is mature and efficient, so the integration of developed pieces of OSS is the best choice to research and develop starting from a solid base point.

C. Open Source Software Reuse

Open Source solutions in educational environment cover standard company needs. Some software has the suitable features and tools for many didactic situations.

The large amount of open source software makes the reuse practices the quickest way to develop and add the needed feature for some specific needs (for example with the sloodle plugin, you can integrate Moodle with Second Life (Simulation Linked Object Oriented Dynamic Learning Environment, 2009).
Even in case of Open Source Software Reuse, apparently, the adoption of OSS code seems always advantageous. In reality the costs are hidden; in addition to the adaptation of portions of code (that could be more or less complex), organizations have to include cost for the selection of modules to be integrated and for updating and maintenance.

Software reuse possibilities open up on three axes: what to reuse, how to reuse it, and where to reuse it. Movement along these three axes increases the breadth of software reuse opportunities in any development effort. Source code’s availability lets the community perpetually improve, fix, and support the reused elements. In some cases, by incorporating the source code of a reused element into the system being built, developers can achieve tight integration and a system can be maintained as a whole (Spinnellis & Szyperski, 2004).

However, as educational services are becoming more important in the software sector (both for OSS and proprietary software), the perceived quality concerns the quality of the educational offered services and methodologies, not necessarily the quality of the software. However, in OSS there is an upside most often not present in proprietary software: anyone can verify the quality of the software code because the source code is available for everyone (Pykalainen, 2007).

On the other hand, high degree of dependence on libraries and external portions of code reflects the problem of dependence on the final product. It means that it need to bring the same libraries and external portion of code to let the software work perfectly.

In addition, the reuse of open source software can generate “isolation.” The problem of isolation is reflected by slow bug fixes and the risk of including in the final product unnecessary portions of dead code.

Those risks may affect final product quality with the inevitable economical impact. A possible solution may be the API (application programming interfaces) approach that allows a good level of abstraction between software layers (Madannmohan & De’, 2004).

In the specific case shown some of the described problems are solved. First of all the most important topic is to choose the software to reuse. A good community of users and developers, great maturity, and commercial use experience are important variables that could influence the choice. All the tools (re)used and integrated are all written in Python language and this helps a lot in developing new compatible modules for the whole system. All technical specs are Open so everyone could develop his own extension.

The LIS module is developed as external application using SOA – SOAP architecture with a public interface through the use of Web Services. The services will be described through WSDL standard and the messages will be coded according SOAP standard.

The use of standards for the services exposure allows interoperability of LIS with all e-Learning platforms and with the whole framework system.

According to this analysis, the choice done is (again) Open Source oriented.

D. OSS Promote Research

As stated in literature, the OSS phenomenon has promoted research (Krogh, & Spaeth, 2007). Everyone knows that the educational field is a permanent research laboratory and, when it is supported by computers (CSCL), the challenge is to find the right mix between didactic methodology (e.g., collaborative learning, collectivism, connectivism, etc.) and technological resources. Open source stimulates viral innovation. OSS models have changed development processes. “Sprint” development approach is rising speed and quality of code (A sprint is a time-boxed period of software development focused on a given list of goals. Sprints have become popular events among some Open Source projects). The most important benefit of sprints is allowing people to meet and collaborate in person (Goth, 2007). OSS pushed proprietary license models to their limits, until market pressures spawned entire new models such as packaging or value-added services.

The framework proposed is a research project and is continuously under development. Open Source model is the best choice for this kind of work.

E. Migration Issues

A possible cost that every company should consider concerns the migration. A company needs to evaluate every situation separately. Migration from Open Software to another Open Software will be quite transparent with no great cost in training or in hardware adjustment (open software are often standard compliant). Migration from proprietary software to an open one could generate relevant costs in training for total renewal of technical know how and to rebuild some contents according to new software specifications.

An organization that would adopt OSS as educational tool has to consider the quality of Open Source Software that will be implemented. This topic is at the center of last research debate because the quality concept is quite abstract and it depends on some factors:

Security: Intrinsic quality of software is the transparency of source code. Everyone could correct bugs or find malware.

Development and support community: A software project is evaluated as mature considering the following indicators: quantity and quality of documentation, frequency of the release, efficiency of the bug report system and speed in bugfix.

The high degree of maturity of a project often creates commercial sponsorship with companies involved in promoting and leading the development of the project and offer commercial support and consulting (e. g., docebo or Moodle).

Many companies nowadays have adopted the OSS system integrator business, consisting in OSS component integration covering OSS lacks (such as support, brand, organization and verticalization of software according to specific needs).

XIV. CONCLUSIONS

As showed in this experience of research and development of innovative AI-e-learning supported project, it is clear that everyone can assert that the adoption of OSS software in educational environment is not free of cost or threats.

Even if there is a great probability of an innovative product (with high investments on R&D), an organization has to deeply analyze internal (e.g., ideological and technical issues) and external factors (e.g., software
selection) before choosing its own educational tools and the business model suitable to supporting it.

Despite the identification of factors and variables that contribute to do a deep evaluation about this choice, it is not possible to define a model of characterization that can help to address a sensitive junction since the empirical data collected is not enough to demonstrate the theory.

In this field, the research groups are trying to define a model that could harmonize the largest number of variables, in order to give a valid aid to corporate management.

REFERENCES

[11] Learning Environments for Progressive Inquiry Research Group - UAH Media Lab, University of Art and Design Helsinki - In cooperation with Centre for Research on Networked Learning and Knowledge Building, Department of Psychology, University of Helsinki. (2006). http://fle3.uah.fi/ (Not cited in the paper unless the one cited on page 9. This URL takes you to a page titled Fle3 > Future Learning Environment. If this is the work you are citing, the title should be shown here.) This is the link that explain who’s the Fle3 team

AUTHORS

V. L. Plantamura is Full Professor of Fundamental Computer Science at the Faculty of Mathematical, Physical and Natural Science of the University of Bari since 1975, area INF/01. He was charged with several teaching assignments such as Programming and Information System.

Dean of the Degree in Computer Science and Digital Communication at the Faculty of Mathematical, Physical and Natural Science of the University of Bari; Chief-coordinator of the “University Bechelor in Computer Science” at the Faculty of Mathematical, Physical and Natural Science of the University of Bari; University of Bari Rector’s Delegate for Technology innovation;

The research interests and the relative activities have always been focused on analyses and implementation of computer science systems with particular reference to their performance evaluation. Presently, his research is involved in defining learning models for the project of engineering training paths.

(e-mail plantamura@di.uniba.it)

A. Marengo Assistant Professor in Faculty of Economics at University of Bari. His research activity takes place primarily on didactic methodologies implemented by the use of ICT tools, particularly the development of e-learning web-based platforms that compete to introduce the technologies of distance learning in traditional institutional campus courses and activities.

(e-mail: marengo@di.uniba.it)

A. Pagano is a Phd student in Computer Science at University of Bari. His research topic is about Open Source impact on Enterprise Software Development. He was a researcher for the “research 60%” focused on “planning methods and implementation of didactic models and artifacts, focused on learner and focused on knowledge construction communities. He is ICT Manager for Osel Consulting srl – SpinOff of University of Bari. He is a LUGBari member (Linux User Group). He is an Open Source philosophy supporter

(e-mail: alessandropagano@di.uniba.it).

Manuscript received 31 March 2010.

This work was supported in part by the U.S. Department of Commerce under Grant BSI23456 (sponsor and financial support acknowledgment goes here).