Remote Cooperation on Project-centred Learning: a Working Implemented Solution in Academia

Carola Salvioni[1], Antonio Vincenzo Taddeo[2]

[1]Faculty of Communication Sciences, and [2]ALaRI, Faculty of Informatics,
University of Lugano, via G. Buffi, 13, Lugano, Switzerland
carola.salvioni@lu.unisi.ch, antonio.taddeo@alari.ch

Abstract. The paper aims at illustrating the original technical solution provided within an academic institute in order to manage teaching activities, encompassing the coordination of project-centred learning processes that run in parallel with the formal theoretical courses.

Unlike the planning of the academic teaching that can be scheduled year by year, the development of a project cannot be defined over a long period, but it requires frequent report reviews and updating by the different actors involved in the project. From this consideration, and due to the peculiar context of the ALaRI institute, it was clear the necessity to manage asynchronous and synchronous communications occurring during the ongoing project, facilitating the team members' remote interactions and cooperation.

The provided solution within the EU COOPER project is the answer to more and more common scenarios of use, reflecting not only university requirements, but also industrial needs based on the cooperative teamwork among persons geographically dispersed and with heterogeneous competences.

1. Introduction

At present, the academic environment is more and more characterized by three peculiar conditions: *i)* the high mobility of students, lecturers and workforce, who can belong to the same institution or – more frequently – come from different organizations to work together; *ii)* the setting up of "classes" composed of persons with different competencies, backgrounds, and culture; *iii)* the increasingly *project-centred* teaching approach. In fact, especially in the context of workplace learning, scientific and technical studies are more and more based on concrete practices and real-life issues, and not only on theoretical and formal lectures. In this situation teams rather than individual students work on a given project, in which teachers' support is often substituted by interaction with a small group of advisors and tutors, possibly coming from different (remote) institutions and providing different competencies and approaches. This teaching perspective is considered as a valuable pedagogical addition for developing the multidisciplinary attitude of the learner.

This paper reports the empirical experience of an academic institute that has adopted such a teaching approach based on the project-centred learning.

Consequently, the institute has faced a set of demands that brought it to adopt a specific technical solution for managing its internal requirements, in order to support and to enhance its complex network of human communicative interactions, ensuring data security and confidentiality of private information through a secure and fully operative workplace.

In the following sections, the reader will discover the conditions and the reasons that have led to designing, building and using specific services supported by a technical platform to satisfy such new educational needs. In particular the section 2 illustrates the academic environment, its learning solutions, and the social actors; the section 3 underlines the tackled problems in such a scenario; while how these problems have been faced up through the implemented technology follows in section 4. Finally a preliminary evaluation of the working solution concludes this user case in section 5.

2. The Academic Scenario

ALaRI[1] is the institute at the University of Lugano[2], Switzerland, promoting research and education in the embedded systems field since 1999.

This institute is the outcome of the synergic interaction between prestigious universities and international high-tech industries with the aim of fostering academic-industrial collaboration in the field of the embedded systems. In this perspective, ALaRI offers a cross-disciplinary education, through design laboratories and projects, involving fields from electronic engineering to computer science, as well as the development of personal skills, such as team work and complex-project management. The research topics are based on real life design issues, generated by actual technical and electronic industrial requirements, and mainly focused on security of mobile systems, pervasive computing and low-power design.

Within this framework, ALaRI's challenging goal is to prepare high level profiles in embedded system design through its educational opportunities. In particular, at present, the institute offers two master's programs in embedded systems design, differently tailored and open to international participants from all over the world, having a technological or scientific background.

The Master of Advanced Studies (MAS) is a one-year program, for postgraduates or candidates with work experience (this master can be attended also in a part-time solution, split over two years); then, since 2004, the Master of Science (MSc) presents a two-year program complying with the latest European guidelines, and intended for students with a Bachelor degree. Within this master program there are two tracks of study: *Design and Research* (oriented to the design and research activity in academic or industrial environment), and *Business Projects* (for the management, economics and marketing of embedded systems).

[1] Advanced Learning and Research Institute (ALaRI) – www.alari.ch

[2] The Università della Svizzera italiana (University of Lugano), active since 1996, is the youngest among the ten Cantonal Swiss Universities and the two Federal Technical Universities belonging to the Swiss Higher Education System and recognized by the Swiss Federal Government – www.unisi.ch

The teaching staff of the institute relies on about thirty international lecturers, coming from prestigious EU and US academia and research institutes, as well as experts from high-tech multinational companies.

Through a quite unique educational program, the institute offers an innovative approach to the working organization and learning environment, combining learning in presence with *remote tutoring*. In particular, two main activities, partially overlapping, are developed during the master's programs: the courses' lessons and the development of master's research projects, according to the project-centred learning approach. While the courses are held according to the classical face-to-face learning fashion, with lecturers and students in presence; the projects are developed through a *remote tutoring* requiring a *remote cooperation* among all the team members. In fact, students, physically working at the institute need to interact with their project supervisors (lecturers and industrial experts), who, on the contrary, are mostly geographically dispersed.

During the academic year, the learning is organized into teaching units (also called "modules"), whose length may range from 12 to 36 hours, including theory, exercises and practice. Modules end with an individual evaluation that may include assignments or a module project. With very few exceptions, lecturers are present at the institute in Lugano only during their period of teaching that normally is distributed over 1 or 2 weeks. This means that both master's programs consist of very intensive cycles of lessons concentrated during the lecturers' short physical presence.

Further, master research projects run in parallel with conventional studies and are integral part of the programs, completing the students' training, and leading to the final master thesis (i.e. the project outcome constituting the final dissertation). The applied-research projects are normally suggested by the industrial sponsors or collaborators of the institute and concern actual industrial research, design activities and technological needs. The projects are assigned to each student early in the academic year (in the first academic year for the MSc students), and checked periodically by lecturers and industrial experts through remote interactions. Then, several parallel projects may complete a larger research activity, where practical experience in teamwork allows participants to grasp the problems of design management from the perspective of work organization as well as financial relations.

Thus, during the master's programs students are trained to work both on their own (and in team work) and interacting remotely with their supervisors tutoring the development of the projects. Normally there are 3 figures covering the roles of supervisors: a lecturer or academic member who provides scientific counsel for the project; an industrial experts (sponsor of the project or collaborator of the institute) who offers the industrial guidelines about the project topic and expects some results from it; and a tutor, who often is a phd students belonging to the ALaRI staff, in charge of supervising projects of his competence. The tutor is, among the three supervisors, the closer to the student(s) in everyday work.

This situation requires, first of all, a great effort of coordination and synergic interactions among all the social actors involved in the institute, who can be briefly grouped in the following profiles listed in table 1.

Table 1. *ALaRI social actors' profiles*

Actor	Role
Students	Students attending every academic year one of the two master's programs
Faculty	Lecturers and experts who hold courses and in some cases provide academic supervision for master projects.
Sponsors and Industrial collaborators	Industrial partners, supporting scholarships, and collaborators who supervise master projects.
ALaRI Staff	PhD students, with supervision role on master projects, and other administrative staff members normally located at the institute.
Scientific Council (or steering committee)	Selected faculty members who direct the scientific strategies of the institute and remotely supervise the research activity at the institute.
Alumni	Graduated students maintain contacts with the institute and access teaching materials and their research projects for a few years after their graduation.
Guests	Anyone outside the ALaRI network, interested in accessing public information of the institute.

3. Tackled Problems

The particular teaching program characterized by face-to-face lessons but with a remote tutoring risked of compromising the quality of the relationships not only between the faculty and students, due to the lecturers' very short presence at the institute, but also among the other social actors geographically dispersed who need to interact remotely to define the various educational activities.

Further, initially, personnel dedicated to the general administration of the institute was missing, and only the program manager of the institute was totally (full-time) devoted to the management of all the administrative and learning issues. This entailed a huge traffic and a sustained exchange of messages, emails, and phone calls overwhelming the program manager's activities. Thus, from the beginning, this situation has required a great organizational effort in order to coordinate both the annual academic teaching plan and the remote work for the organization of the course's schedule and for the development of the master's research projects.

At a general level, it was necessary to tackle two main problems:

- one was about the management of well-defined workflow procedures whose outputs consist of static information that can be fixed and stored year by year in a suitable repository in order to find it quickly or to update it in case of need
- the other was about the managing of remote cooperation processes that can be only partially defined in advance, but never completely specified until runtime, as analysed in [1]. Further, it was necessary to keep track of the various cooperation processes results, managing dynamic and brief-length information that evolves frequently in a period ranging from two weeks to three months.

In the specific, we refer in the first case to the general administration and the course's organization, before and after the lessons in presence. For instance, it is necessary to schedule the courses according to each lecturer's commitment; or lecturers need to distribute their teaching material; while students need to access it; then, credits and final marks must be assigned, after the remote evaluation of students' exams; and so on.

Moreover, the increasing number of persons (students, staff, lecturers and industrial experts) around the institute led to the necessity of quickly adding, retrieving, modifying and updating the social actors' personal data and their relative administrative documentations.

In the second case, we especially refer to the support for remote cooperation processes tutoring students working on complex projects. In fact, in the ALaRI institute, the project-centred learning is a fundamental part of the master's programs. The project follows a sequence of phases constituting the project life-cycle. Some of these phases have well-defined and structured processes; some others have unstructured set of procedures which cannot be arranged in a pre-defined way at design time. Each project is developed by one team whose members cover different roles, having different competences and background (e.g. students, tutors, teachers, industrial experts), and, in our specific case, are also geographically dispersed. In fact, the different actors revolving around the project need to interact also remotely to participate in the completion of the research work. For instance, students need to expose some results during the project evolution, while teacher or industrial expert need to evaluated them. For this reason, the team members need tools supporting both synchronous and asynchronous communication. Then, some project resources are available at the beginning of the project, but most of them are created by the project's members when the project is still ongoing. This condition requires keeping track of these developments, in order to maintain the know-how and expertise on the project, until the final release of the achieved results. Further, the project resources can be public (like the final report) or private (like some documents produced during the evolution of the project) and accessible only by the persons strictly involved in the same project and by the ALaRI scientific council.

Another not irrelevant problem was about to fix a common infrastructure to the project management, since each team with its members geographically dispersed managed the project following own practices and methods. In fact the lack of a common practice made difficult not only to the ALaRI scientific council the control and supervision of the various projects evolution during the academic year, but also at the end the collection and presentation of the various project's results in compliance with a recognized common style.

Then, the achieved project's results and the increasing educational materials required creating a knowledge base repository with the aim of supporting the knowledge co-construction, sharing and re-use, exploiting the results obtained from the completed researches. In particular, the retrieval of the project outcomes constitutes a worthwhile support to enhance not only the individual learning, but also the knowledge property of the institute, making it independent from the individual know-how, and reducing the leak of intellectual property due to the physiologic turnover of human resources in such an academic environment.

In particular, the management of project's data and information opened another issue about the security of the information system. In fact it was necessary to develop an access rights mechanism to protect the company know-how and manage permissions among the different actors.

It is easy to understand that from such a complex scenario, it was very urgent to find a suitable solution. Minimizing the direct relations, it became necessary promoting the high quality of the remote interactions, through synchronous and asynchronous communications, ensuring interactive participation of all the social actors and facilitating their remote cooperation with respect to training, team working as well as to all the phases of work planning.

4. The Provided Solution

The answer to tackle such problems was the building from scratch of the ALaRI intranet: a web-based remote application accessible from the ALaRI web site – http://www.ALaRI.ch/intranet. It was designed using a high-level modeling language, called WebML [2], and then deployed using a WebML design tool and an automatic code generator, Webratio [3]. Through this application, the platform consists of i) a data model representing the entity-relationship diagram of the used platform database; ii) an hypertext model describing how the web pages are linked together and the content structure of such pages; iii) and a presentation model used to define the web page layout positioning its contents and graphical presentation. Such a web application designed with WebML is divided into siteviews, which represent a particular, filtered view of the underlying data structures, organized into pages and areas (collections of related pages).

The ALaRI platform development can be divided in two main periods. The first one occurred within the EU project ANTITESYS (2002-2004) [4] [5]; while the second is still actually ongoing within the EU project COOPER (2005-2007) [6]. The study of the scenario requirements led to implementing, in the first phase, a platform supporting some fundamental services that allowed better managing the organization of administrative and teaching activities, especially supporting the remote tutoring before and after the lessons in presence. These services can be roughly grouped in the following macro-area:

- *People Directory*, where each member can easily manage and update his/her personal profile, such as a public CV (filling standard form) including contacts and personal links accessible by other community members.
- *Courses*, supporting remote tutoring and local teaching, where lecturers and students can easily interact in an asynchronous way. For instance, lecturers can upload from remote the teaching material in electronic format; while students can consult it during, before (for preparation) and after (for assignment) the course; then, each student has visibility on the own learning page where credits and marks are assigned by the lecturers after the evaluation of students' exam.
- *Master Project*, characterized by a milestone/report interaction model where supervisors can set milestones and students respond uploading the respective documents.
- *Knowledge Repository*, to collect and keep the company know-how, taking advantage of the knowledge sharing, quickly retrieving and re-using the stored

static documents, such as policies, master theses, final project results

The originality of this first solution consisted in integrating the various services into a unique platform, with *multi-directional navigational* patterns, but limited by an *advanced data filtering* system based on user type and status to guarantees the security of the uploaded information. [5].

Actually, on the platform there are other interesting services and functionalities, but their description goes beyond the purpose of this paper. On the contrary, it is important to underline that not a few problems remained in order to managing the remote cooperation processes for tutoring the research projects development, since the primitive provided service only partially covered the various phases of the project-centred learning approach.

Therefore, specifically within the COOPER project, it was carried out a further detailed study, also comparing other similar scenarios both in academic and industrial environment, focusing on the definition of a reference framework for the team management processes [7]. In particular, there was defined the principal project team-work processes [1], occurring during the project-life cycle. Thanks to these studies, it was possible to improve the management of the remote cooperation for the research project on the ALaRI platform. Further, also the area dedicated to the knowledge repository has been improved adding features supporting the virtual building of the project bibliography. How these features have been designed is reported here below.

Design of the business processes in a project life-cycle

Analysing the requirements of the ALaRI institute in terms of project's organization and project's processes, we have recognized four main project phases constituting the project-centred sections on the ALaRI platform. These project life-cycle phases are composed of: *Project Definition, Project Assignment, Project Development* and *Results Submission* (in time sequence order). The first two steps represent pre-project phases; while the last one is related to the post-project phase.

In particular, each phase includes two different types of processes: static (structured) processes and dynamic (unstructured) processes [7] [1]. Both processes may involve a single user or different actors. An important aspect of the above schema is that the *Execution* phase is characterized mainly through dynamic processes; instead the other phases are based on well defined procedures, usually performed in a chronological sequence. The ALaRI platform implements such processes (both static and dynamic ones) through several web pages and special components developed to fulfil its requirements. A detailed description of the processes implementation follows together with the definition of static and dynamic processes.

The *static processes* can be envisioned and specified at design time defining precisely flow, synchronizations, constraints and actors. On the ALaRI platform, for instance, they were used to define the project assignment process shown in Figure 1.

Here the project assignment phase has been modelled as a business process in order to identify the main actors' interactions and executed activities. In Figure 1, one can notice the involved actors (the lanes), the operations performed (the bubbles), their interactions (the normal arrows) and the resources created/exchanged (the boxes and dashed arrows). Some of these activities can be executed in parallel (e.g. the evaluation phase in figure).

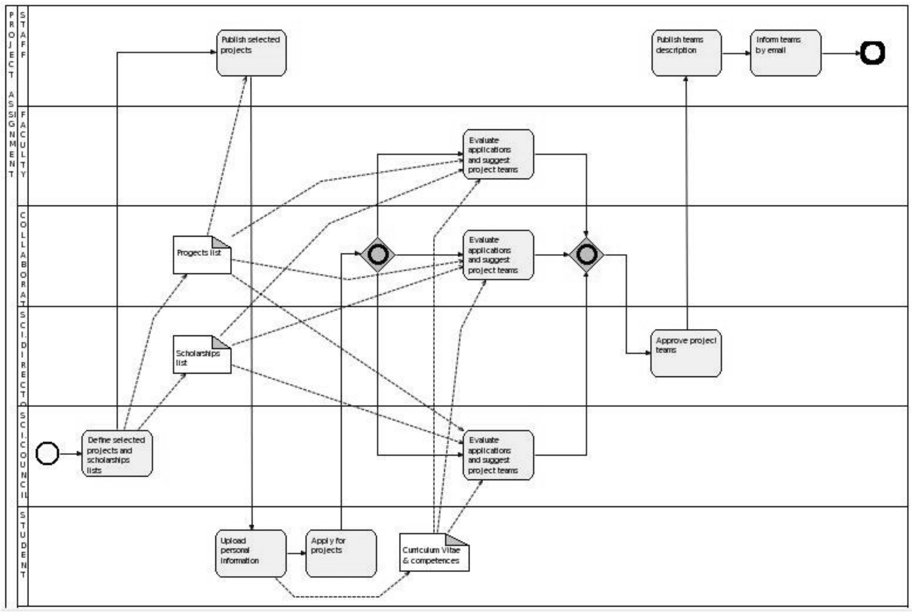

Fig. 1. ALaRI Project Assignment phase

The static processes have been specified using the BPMN notation [9]. They are then automatically translated, using the BPMN editor, from a workflow into a skeleton of WebML hypertext diagram in order to be easily customizable by the web application designer. In this way, we have been reduced drastically the design time.

The *dynamic processes* definition and creation are presented in depth in [8]. Basically, they consist of a set of typical and predictable activities for which the independent behaviour as *atomic actions* is well-known [1]. A library of *atomic actions* is a set of pre-coded activities that the user can either invoke as stand-alone actions or compose to structure dynamic processes. Process composition proceeds by selecting activities within the library of atomic actions, actors, resources and execution constraints and, finally, organizing them in different steps. However, possible composition of such activities and the assignment to actors are not easy to predict at design time, since they may depend on users' choices at enactment-time. Cooperation processes are thus dynamic and user-driven.

In our case, we used such a concept to build a set of *atomic actions* which can be combined together to build, dynamically on user demand, a sequence of operations establishing a process. Atomic actions can be used for managing resources (e.g. publishing a resource, acquiring or routing it, etc.), or teamwork (e.g. assigning roles or tasks, defining tasks, milestone, or deliverable, etc.).

The definition of a dynamic process can occur through the web interface that drives the user in the composition of an atomic actions sequence. In particular, the definition of a process phase requires the user to i) select one activity or a group of activity from the library; eventually defining one kind of activity parallelism; ii) assign each of the selected activities to a team member, in case of group activity the user is also required

to select one kind of parallelism; iii) specify possible resources that are the objects on which process activities operate. Such resources can be also shared by different users and along different activities.
If a process is composed of multiple steps, than the above operations must be iterated until all steps are covered.

Knowledge Repository
The Knowledge Repository (KB) system has been developed in order to allow the user i) to *upload* documents, characterizing with metadata such as name, abstract, representative keywords, authors and also personal comments, in a common shared repository; ii) to *search* the repository for documents already uploaded by other users; iii) to *organize* all the interesting documents into private virtual folders; iv) to *build* the own project bibliography by moving project-related documents into a special bibliography folder, shared with the other team members.

The developed sharing model could be built upon the concept of *virtual folders*, that are characterized by *flexible semantics* - virtual folders could represent a logical collection of documents referring either to a particular topic on which the user is interested on (e.g. “hardware”, “compilers” cryptography”, etc...) or to a specific task that s/he intends to perform on the set of documents (e.g. “read”, “review”, etc...); and *virtuality* - each folder is not properly a collection of files, but rather a collection of links to files. This allows virtual folders (even if created by different users) to overlap with each other by linking to the same documents, thus avoiding the duplication of files in the repository, saving space and, at the same time, allowing a perfect implementation of the knowledge sharing requirement. A virtual folder represents a particular view on the whole repository.
Furthermore, the KB provides multi-directional navigation capabilities among authors, documents and folders. For instance, the user might start browsing the KB, find one interesting document, get details on it; see its owner, or its authors profiles; move on to the related projects; get reports about those projects, see which professor supervised them, and so on.

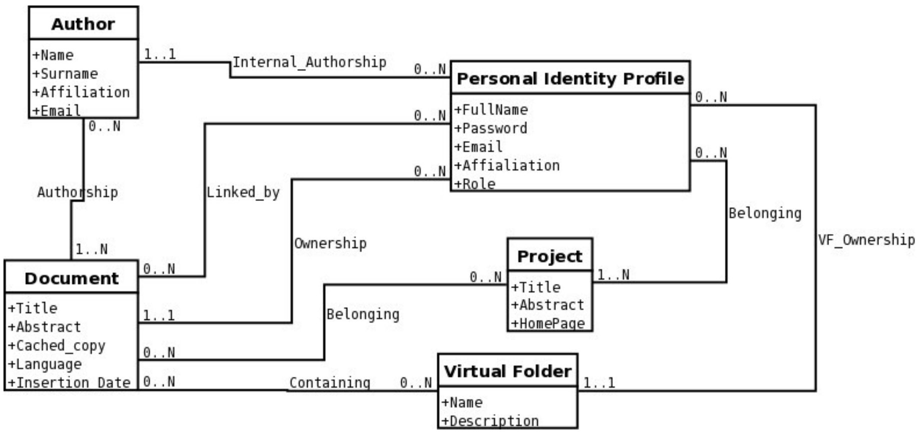

Fig. 2. Knowledge Repository Data Model

The KB is composed by a data model and a navigational model. The data model of the KB is shown in Figure 2. It is a simplified representation of the Entity-Relationship diagram, in which the main entities are drawn, with their minimum set of attributes. Apart the classical relationships between *Author*, *Document* and *Personal Identity Profile* (user), there are two specific ones: the *Linked_by* and the *Ownership.* The first takes into account the number of users which have "linked" a given document into their virtual folders (described later). Such a relationship represents a sort of "ranking" of a given document. The higher is the number of "links" to a document, the higher will be its evaluation from users' perspective. The latter, the *Ownership* relationship, is added to store the information regarding the owner of a document which has some access rights on it (e.g. the owner can modify and/or delete his uploaded documents).

Furthermore, a document may belong to a project; in this case a record is added to the relationship among *Project* and *Document*.

Finally, the virtual folders implementation is achieved through *VirtualFolder* entity, belonging to a user, and that can be linked with several other documents.

In addition, we developed an advanced Role Based Access Control (RBAC) mechanism to manage the user's access rights not only for the Knowledge Repository but also for the projects.

5. Preliminary evaluation of the implemented platform

In order to perform a rough evaluation we have added some monitoring units (e.g. activity log based) through which we have been able to capture some usage of our platform. The main idea is to verify if the implementation of the main business processes into a web application allows increasing the company efficiency and in particular the actors' cooperation.

As preliminary results, we have observed that the platform is mainly used to share resources (e.g. documents) and to setup implemented synchronous communication [10] [11]; while other services, such as searching into the repository, are less used; we can expected that the repository usage will increase with the increasing number of documents. This consideration proves that the cooperation among actors has gained advantages from the platform.

Another important advantage from the platform improvement has been the management of the administrative procedures which has been done in a more efficient way. The implementation of the static processes has helped the staff members, collaborating with the program manager, to organize the institute activities, for instance providing a global vision on "who is doing what". Moreover, the use of static procedures has entailed the introduction of precise policies, providing the geographically dispersed actors with a common infrastructure to the project management.

Future improvements will be devoted towards the usability and customization of the web interface. In fact, some difficulties have been met at different levels especially considering semiotic, cognitive, and navigational issues, as reported in a usability internal report for the ALaRI platform. For instance, the identification of meaningful links on the web page has led to working on reducing at minimum the number of mouse-clicks to perform the most common user's operations, by improving the design of processes and single actions inside site-views, areas and pages.

Acknowledgements. This work is partially supported by the EU/IST FP6 STREP project COOPER (contract number IST-2005-027073). We are really grateful to all cooper members for their collaboration in the effort of improving the processes of and services for the project-centred learning approach, experimented within the ALaRI environment.

References

1. S. Ceri, M. Matera, A. Raffo, H. Spoelstra. *Flexible Processes in Project-Centred Learning*. Proc. of EC-TEL'07, Crete, Greece, September 2007 (in print).
2. S. Ceri, P. Fraternali, A. Bongio: *Web Modeling Language (WebML): a modeling language for designing Web sites*. Computer Networks 33(1-6): 137-157 (2000)
3. Webratio (actually WebModels), more information available at http://www.webratio.com
4. Antitesys: more information available at http://antitesys.alari.ch
5. L. Negri & U. Bondi, *The ALaRI Intranet: a Remote Collaboration Platform for a Worldwide Learning and Research Network*, ED-Media, Lugano, July 2004
6. COOPER: more information available at http://cooper-project.org
7. A. Bongio, J. van Bruggen, S. Ceri, V. Cristea, P. Dolog, A. Hoffmann, M. Matera, M. Mura, A. Taddeo, X. Zhou, L. Zoni. *"COOPER: Towards A Collaborative Open Environment of Project-centred Learning"* in EC-TEL'06 conference proceedings, Crete, Greece - October 1-4, 2006
8. Brambilla, M., Ceri, S., Fraternali, P., and Manolescu, I. 2006. "Process modeling in Web applications". *ACM Trans. Softw. Eng. Methodol.* 15, 4 (Oct. 2006), 360-409.
9. M. Brambilla. "Generation of WebML Web Application Models". ICWE2006, Menlo Park, California, USA.
10. Web Models, Abbeynet, "WebRatio 4.COOPER , Guide to New Features" , October 2006, COOPER deliverable D4.2.
11. N. Aste, A. Bongio, S. Butti, S. De Marco, L. Filigheddu. *Specification and Design of Process Modeling Concepts and of VOI Units in WebRatio.* Cooper Tech. Rep. http://www.cooper-project.org/public_docs.html.