

Design Issues in Developing a Learning Environment For Small-sized Enterprises Communities

Fiorella De Cindio
Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
fiorella.decindio@unimi.it

Ivan Longhi
Rete Civica di Milano
ivan.longhi@rcm.inet.it

Laura Anna Ripamonti
Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
ripamonti@dico.unimi.it

ABSTRACT

This paper investigates the technological issues involved in the design and implementation of a Web environment intended to support learning processes in small-sized enterprises communities.

Actually, since small and medium sized enterprises (SME) enter interfirm relationships for they realize immediately their actual and future benefits, Internet infrastructure and services seem as a perfect fit to SME's networks (they seem accessible at moderate cost, available and flexible), for any firm may contribute to the nature of the information-based exchange. Thus, a large number of small cooperating firms should be ready to use such a technology. Instead, empirical evidence suggests that adoption and use of ICT (Information and Telecommunication Technologies) and Internet by SME in Italy is making small progress. One of the main reasons of such misalignment is that SME do not "trust" the net (for it still looks like an expensive highway whose real benefit still remains unclear), nor understand completely the potentiality offered by the technology to the business, in term of effectiveness and support to their processes.

To address this problem the "Partecipate: Just do it!" project has been started, creating an integrated web environment as a learning tool for SME. The overall technological infrastructure supporting this environment is quite complex, and a specific Learning Management System has been developed. This work tries to outline the LMS architecture and choices that have been made during its development.

1. JDI! as a learning tool

The “Partecipate: Just do it!” project was started in spring 2002. Its main goal is to make up for the SMEs’ lack of information/skills in ICT - Information and Communication Technologies (among the main reasons for SMEs’ mistrust toward Internet use) and to allow them to ‘experience’ ICT’s potential benefits for their business. Thus a ‘protected’ Web based community environment has been created, expressly developed for SME needs and free from ‘business interests’, in order to make them feel confident about the quality and honesty of the services and information supplied. Actually, since small and medium sized enterprises (SME) enter interfirm relationships for they realize immediately their actual and future benefits, Internet infrastructure and services seem as a perfect fit to SME’s networks (they seem accessible at moderate cost, available and flexible), for any firm may contribute to the nature of the information-based exchange. Thus, a large number of small cooperating firms should be ready to use such a technology. Instead, empirical evidence suggests that adoption and use of ICT and the Internet by SME is making small progresses [BMM02], [BDR04].

One of the main reasons of such misalignment is that SME do not “trust” the net (for it still looks like an expensive highway whose real benefit still remains unclear), nor understand completely the potentiality offered by the technology to the business, in term of effectiveness and support to their processes [But02].

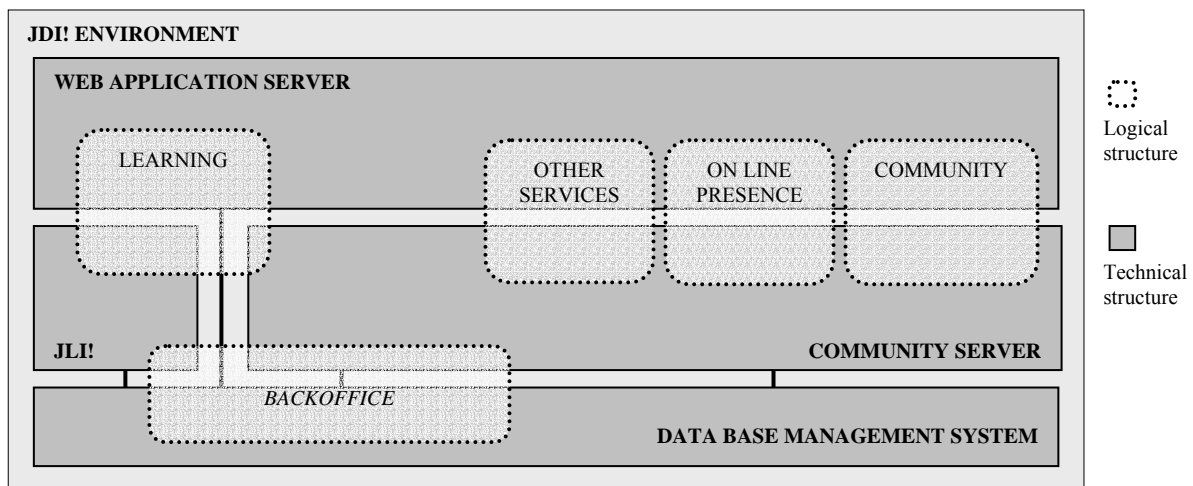


Fig.1 – “Just do it!” logical and technical infrastructure (main features)

To address this problem the “Partecipate: Just do it!” (JDI!) project created an integrated web environment as a learning tool for SME [DLR03].

Since the “Partecipate: just do it!” environment couples learning with community tools, while designing it at least three sets of different major requirements – on both the design and implementation levels - required our attention [Teg02], [Kim00]: defining, on a ‘logical’ basis, an integrated, sound, and effective set of services for SMEs; creating a technological infrastructure that could support the designed set of services; and involving SMEs in the project in order to assure that its goals were consistent with SME needs. Here, we will concentrate on the technological infrastructure design aspect, providing only a short overview on the other two issues, that are detailed in [BDR04]. The JDI! environment is made up of several different integrated (yet independent) tools, including:

- **learning services:** JDI! learning services are focused on giving SMEs the basics of Internet and ICT use, in order to enable them to evaluate on their own which possible *real* benefits may derive for their business from technologies. Since empirical evidence demonstrates that one of the major barriers in adopting net technologies (and ICT more in general) derives from the fact that education on ICT and technology supply is generally targeted at large companies, we decided to offer SME, alongside with basic courses, the possibility to propose the activation of specific courses, based on their real information needs. Moreover, since we think a learning community is the critical factor to stress for making SMEs feel confident in the Internet as a mean for co-operation, we integrated

community functionalities with learning services, creating forums for each course, aimed at stimulating interaction between enterprises and teachers, and among enterprises [CDR01], [Wen01], [BrD91], [BrD00].

- community: since both theoretical and empirical evidence [BDR04] demonstrate that the *trust* component could be positively influenced by experience sharing, to enhance collaboration and the development of entrepreneurial networks through learning communities and communities of practice, we provided several community services, such as:
 - “general purpose” forums, where entrepreneurs can interact in a protected environment, in which experienced “moderators” will act as facilitators to support interaction development, and that will be specialized alongside with discussion threads development;
 - forums to propose and discuss new specific courses activation and their priorities; thus entrepreneurs are stimulated to share their experiences and knowledge, learning “hands on” the basics of community of practice/learning communities;
 - “the experts answers...”: forums to discuss specific problems (related to the use of ICT for business) with experts in the field;
 - Internet resources (such as newsgroups, mailing lists, etc.) selected by the JDI! Project team and periodically summarized for providing SMEs with up-to-date technology information;
- “tools for being online”: simple and flexible tools specifically conceived to help enterprises manage their online interactive presence by themselves (e.g. to build intranet/extranet, Web pages, etc.)

2. Technological architecture of JLI!

Due to the JDI! goals, one among the core functionalities of its technological infrastructure is the Learning Management System (LMS). The use of existing solutions for the e-learning platform demonstrated to be impossible, since none of the product we’ve examined (neither commercial, nor open-source) supported such a strong integration with community tools as we needed [LaW91], [Pree00]. Thus, to support the overall structure described in the previous chapter, we’ve decided to adapt and integrate (see Fig.1) several existing/already developed tools and to develop our own e-learning platform (JLI! – Just Learn It!), starting from the Adept (<http://adept.sourceforge.net>) open-source project, even if its latest version was quite less than a “beta release”.

Here in the follow we will briefly describe both the technical and functional features of JLI!. In particular the functional description will include the analysis of users roles, users management, courses management, and learning objects management (Fig.3).

2.1 Technical description

The overall infrastructure of JLI!, has been developed to be an upgradeable open platform, integrable with other “products” in a simple way, to reach the needed interoperability among the different tools used in the JDI! environment. The system is platform-independent, and runs in Apache/PHP environment, using MySQL DBMS.

From a technical point of view, since Adept prototype presented several architectural problems and wasn’t intended to be integrated with other platforms, we needed to redesign the application structure quite completely, changing several features in order to make them “external” (e.g. discussion forums only implement a logon interface between “JLI! - Just learn it!” e-learning platform and the external community application). Changes to the database structure have also been made: to assure an easy and flexible integrability (e.g. through LDAP directory) in the data base layer implementation all the *calls* to user management functions have been isolated into a library.

While designing the JLI! data base (Fig.2) we’ve focused our attention on creating a global architecture supporting an easy integration with external applications without any need of “heavy intervenes”. To achieve this goal the *User* and *Course* tables have been carefully designed. In particular the *User* table:

- does not have any external keys;
- its primary key is the user’s username in the system;
- a field describing the user role in the system (administrator, teacher, tutor, student) has been added: to optimise performances we decided to maintain such field internal to the table, but, if an external User

table is used, it could be easily ungrouped and put in a new table (in 1:1 relation with the external User table);

The *Course* table has been designed using a similar approach: the numeric identifier of the course refers it, and its external keys (e.g. the referral to the discussion forum associated to each course) are *text* type. Even if this choice may reduce the performances, its application is worth, since makes the system more easily integrable with external applications. According to our experience, usually this field will be filled with the URL or parameters that will be passed on to the interface layer towards the external application in order to build the final URL used to *call* a specific functionality.

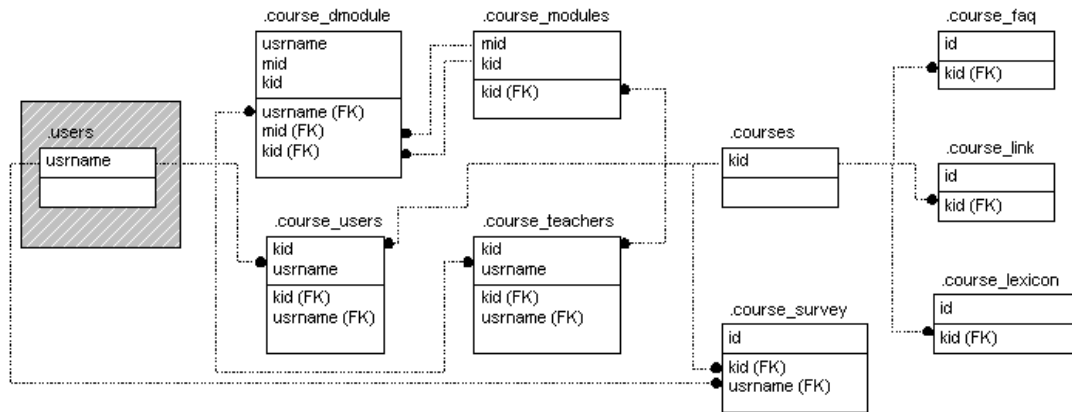


Fig.2 – JLI! data base structure

Specific attention has also been dedicated to the design of the data base supporting *learning objects*. Here our main goal was to obtain a structure as “generic” as possible (Fig.2), in order to make the system compatible with the widest range of learning objects, independently from their structure. In the table containing the data describing the functionalities of the specific learning object (*.course_modules*), is archived also a particular data structure: depending on the learning object type, a specific *routine* will be called and will read this structure to give users (teacher or student) information about the usage modality of each learning object.

In a similar way works the table containing students’ results for each learning object. Depending on the learning object type, a data structure will be created according to the student inputs, and will be opportunely decoded by the system when necessary.

From an implementation point of view, the learning objects management layer has been developed defining a set of *classes* that are instanced during either the creation of the learning objects phase or the fruition phase. Actually the classes do contain exclusively proprieties and methods necessary for defining and evaluating each learning object contents, while the composition of the user interface is delegated to the specific application using the learning objects. Thus the classes could be used outside from JLI!, for example to run surveys.

2.2 Functional description

From a functional point of view, during the design phase, we’ve focused on several features in order to achieve the integration mentioned above. Firstly we needed to create very intuitive and simple interfaces, since users’ (SMEs’) skills in software tools use is often quite low. Also on the teachers’ side we needed to supply a flexible and easy-to-use tool. Actually e-learning platforms usually “publish” ready-to-use courses, while we couldn’t be sure users’ requests for personalized education and training will match with something already existent. As a matter of facts, to supply personalized education and training, it is generally quite impossible to re-use exiting didactical modules, on the contrary, quite surely they will need to be modified, adapted, fragmented, or completely rewritten. Moreover, by its nature, a personalized education could not be built on standards, nor be designed for large number of students; as a consequence, it should be economic and it should not relay on expensive features (e.g. sophisticated graphic layouts). Last but not least, it should allow a very short development time per-course, so to satisfy in reasonable time the educational demand, especially if it concerns the rapidly evolving field of the ICT.

For this reasons, teachers must be able to create new courses (eventually even not respecting commercial standard constraints) in a very short time, with few and low-cost resources (since the critical mass for heavy investments is not assured). In particular we've developed a WYSIWYG interface to create HTML contents immediately usable, and we give teachers the possibility to upload and/or link any other resource he/she may consider useful (e.g. video, documents, etc.) and the possibility to include in the course FAQs and glossaries (whose contents may also be enriched by the direct interaction in the community forums and/or chat among teacher and students).

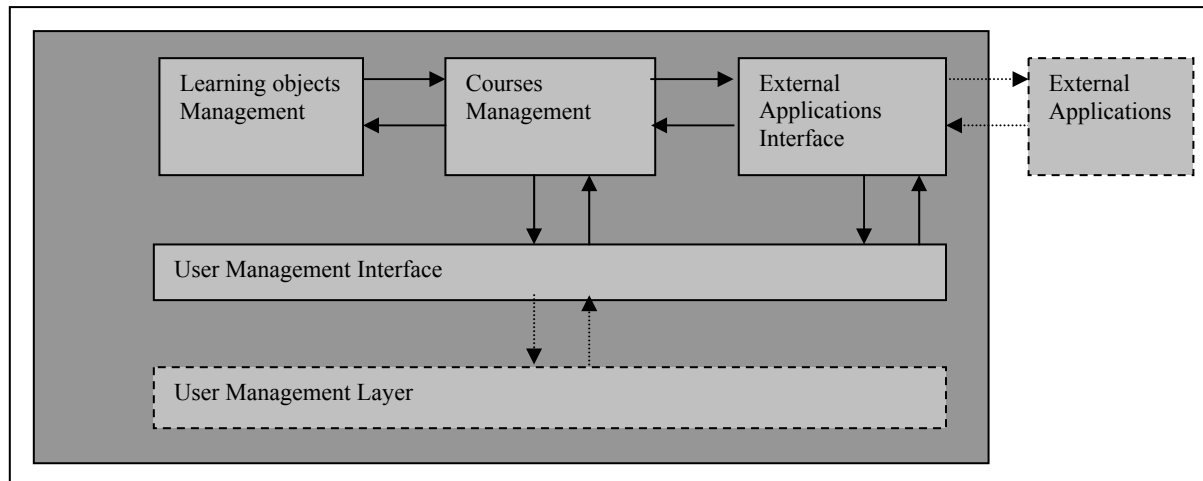


Fig.3 – JLI! system architecture

2.2.1 Users roles

While designing the JLI! access system, four user types have been defined:

- Administrator: manages users and courses
- Teacher: creates, modifies and manages contents and structure of the courses
- Tutor (or facilitator): manages courses carrying out, and the interaction with students
- Student: attends the courses

The *administrator* access to the system grants the privileges necessary to manage users data, courses and *news*. Through the *User Management* interface module the administrator can: see the system users list and their role (administrators, teachers, tutors, students), access and modify the data referred to a specific user, and create new users. By accessing users personal data, s/he can de/assign courses to teachers, tutor and students. Moreover the administrator can also act as a teacher, without the limitation on the directories of students and courses visibility that type of user has.

Accessing the system as a *teacher* will grant the possibility to manage courses, creating a new one or modifying courses assigned by the administrator. Teachers have a restricted visibility on the students and the courses directories: s/he can see his or her courses and the students attending at least one of them. Teachers can access students' information to see which courses they are attending and the results they are obtaining related to a specific course, but s/he cannot modify other users' data.

A *tutor* cannot create or modify courses, since her/his role is to be a "facilitator" for one or more specific courses. A tutor can only see results students are obtaining, correct their exercises where necessary, interact with students using forums, chat, etc.

Last but not least a *student* can access the courses he has subscribed for and several other functionalities offered by JLI!. Once logged on, a student will see a list of courses, that s/he can only access after s/he has selected them or they have been assigned to her/him by an administrator. For each course the number of learning objects s/he has already worked out on the total is shown, and s/he can access a short description of the course. When accessing the course, the student will see a list of the learning objects (and the related state of each one, for example "finished", "waiting for correction", etc.), and links to several functions (course home page, info, FAQ, links, forum). A certificate will be supplied to the student once he has successfully completed all the course's learning objects.

Where possible the student can also choose to subscribe to new courses (or to unsubscribe from uninteresting ones). Finally, a student can also access several other services, some of which are optional and can be activated through configuration variables:

- Results: opens a new window containing the results the student is obtaining in the courses he is attending;
- Notes: to write down notes; it is divided in three different sections – “notes”, “contacts” and “to remember”;
- Glossary: to search specific words teachers are using in courses and for which they have provided a definition;
- Calendar: it is an on line calendar, both personal and of the group. This specific feature uses an application external to JLI!
- Community: through this functionality the student can access a virtual community environment, composed by his virtual class, forums and chats shared with other students and/or teachers and tutors;
- Personal area: personal mailbox, address book, etc.
- Home page management: to manage the student web space. This functionality has not yet been completely developed.

2.2.2 Users management (Interface and Layer)

JLI! has its own users directory that can be managed by the administrator to create, modify, delete, and assign group privileges to users. Nonetheless JLI!, as already stated, has been designed to be easily connected (e.g. using LDAP - Lightweight Directory Access Protocol) to an external users directory, in order to be completely compatible with other applications. In this case the users' directory will be managed by the external server, and JLI! will manage the access authorising on the basis of the external server directives. The same behaviour is expected also by any other server delegated to specific external functionalities, such as: community, forums, chats, calendaring, etc. Thus it could also be possible to implement quite easily new functionalities (e.g. audio and video streaming) in JLI! through the external servers use.

2.2.3 Courses management

Learning objects are the basic didactical unit in JLI!. Learning objects are grouped into *courses* to be supplied to students in a structured way. Each learning object can belong to one or more courses (learning objects can be created ex-novo or can be copied and –if necessary- modified from existing ones. Moreover the system accepts also learning objects formatted with the AICC protocol).

Only *administrators* and *teachers* can create and manage courses. In particular an administrator has access to the whole set of courses present on the JLI! platform at any moment, while a teacher only to the courses s/he created or that s/he had been assigned by the administrator.

A *tutor* can only manage courses he has been assigned to, and a *student* can only attend the courses s/he subscribed to (in particular a student, depending on the system configuration, can attend only courses assigned by the administrator or can choose courses among the ones present in JLI!).

Administrators, teachers and tutors can access the student list of each course. Thus they can monitor the results students are obtaining and the statistical data about actions students do on learning objects. Teachers are provided with the “means” of students' results for each exercise.

When a teacher accesses the data of a specific student with the purpose to monitor her/his results, s/he can also see a log file for each learning object, reporting anything the student had done to it (open, download of multimedia contents, answers saving, file upload, etc.), and whether s/he has finished it or not. Depending on the learning object type, s/he can also access further information and features for the student evaluation:

- S/he can see student answers to multiple choice, true or false, cloze and likert learning objects;
- S/he can download files uploaded by the students
- S/he can correct and evaluate students' answers to the open answer learning objects

Finally, a series of links are present on the interface of the *course management*. They can be used to change or modify course features:

- Info: this link opens a form to input information about the course contents (e.g. short description, requested skills, author, etc.) and to set several conditions. For example the teacher can impose that

learning objects are attended in one specific order or that an email is automatically generated by the system and sent to her/him each time a student finishes the course. Finally teachers can use this interface to make the course become “public” (and thus usable by the students) or “hidden” (for example while the teacher is still working on the course contents);

- Homepage: it is the first page of the course. In this page the teacher is expected to insert useful information for anyone who is beginning to attend the course. To prepare this page, the teacher can either use a WYSIWYG interface or enter directly an HTML page;
- Learning objects management: through this section teachers can create and manage course learning objects;
- Course users: here the teacher can find a section about students results;
- Course evaluation: at the end of the course, each student is asked to fill in a form to give his/her feedback about the course quality to the teacher;
- FAQ: here the teacher can create a list of frequently asked questions and answers about the course contents;
- Links: the teacher can provide a list of useful links related to the course subject;
- Forum: from this link it is possible to access the forum where teacher and student can interact asynchronously;
- Glossary: here the teacher can create a list of useful word.

2.2.4 Learning objects management

JLI! Proposes ten different type of learning objects, divided into three categories: theoretical, evaluation and auto-evaluation learning objects.

JLI! includes three different types of *theoretical* learning objects:

- Theoretical learning object on a single file: it is an HTML page, whose contents can be created both using an on line RTF editor (even using copy & paste functionalities from a .doc or .rtf source file), or inserting text, images and link formatted in HTML. “Preview” and “save” functionalities are also available.
- Theoretical multi-files learning object: in this case the teacher can use the functionalities described above, and can also attach any multi-media content (Words documents, PowerPoint, .PDF, shockwave flash objects, HTML pages, Excel spreadsheets, images, audio, video, etc.) to the learning object. Multi-media contents should be created using external applications and then uploaded through JLI! interface. They will be displayed to students using the browser plug-ins.
- External theoretical learning object: as in the previous case, the reusability of existent contents is stressed. Actually, using this learning object, the student visualizes under the JLI! interface existing web pages or even existing web sites (no matter of their dimension). If necessary, these contents can be protected with a HTTP authentication compatible with the JLI!’s one.

JLI! includes seven types of *evaluation* and *auto-evaluation* learning objects:

- Multiple choice: the teacher creates a set of questions and related answers, signaling to the system the correct ones (answers to each question can also be shuffled each time the exercise is accessed);
- True or false: the teacher inserts a set of questions whose answer is “true” or “false”;
- Cloze: teacher creates sentences leaving blank spaces for the students to fill in with the correct word or words. The system automatically corrects the student answers and can accept more than one correct word for each blank space;
- Likert: a set of sentences is inserted whose correctness degree should be checked by the students;
- Open answer questions: the answer to this type of question is unconstrained. Once students have inserted their answers, the system does not correct the exercise, but let the teacher manually correct it and insert his own suggestions or comments where necessary;
- Hidden answer questions: the purpose of this learning object is to give an auto-evaluation tool to the students. The teacher inserts several questions and answers, but the students could see the correct answers only after he has given its own ones;
- Exercise with a file upload: the teacher will supply the text of an exercise that requests the creation of an electronic document. Once the student has created it, s/he will upload the file for the teacher to evaluate it.

For each of these learning objects types it is possible to upload also images, videos, and audio contents and to link each question to the related theoretical learning object. The teacher can also decide if the student will see the correct answers or only the evaluation.

3. Empirical experience

The JLI! platform is currently used in two different test beds: it supports the courses prepared for the twenty-five SME entrepreneurs involved in the JDI! project, and it is used to perform the examinations of the English courses at DICO – Dipartimento di Informatica e Comunicazione dell'Università degli Studi di Milano. The results are encouraging, and the continuous interaction with users (both teachers and students) enables us to rapidly and flexibly implement new features and modify old ones.

Acknowledgement

The “Just do it!” project is partially founded by the Cariplo Foundation, and is carried on by the RCM Foundation, with the collaboration of Formaper -special agency of the Milan Commerce Chamber- and Iriscube. The authors wish to thank Philip Grew and Aurora Ghini that patiently tested our JLI! platform, giving us useful feedback, derived from their many years teaching experience.

References

- [BDR04] Benassi M., De Cindio F., Ripamonti L.A., *Network of small firms and the web: looking for a trustworthy architect* Paper submitted to Hawaii International Conference on System Sciences (HICSS), January, 5-8, 2004, Big Island, Hawaii
- [BrD91] Brown, J. S., & Duguid, P. *Organizational learning and communities-of-practice: Toward a unified view of working, learning and innovation*. Organization Science, 2, pp: 40-57, 1991
- [BrD00] Brown, J. S. & Duguid, P. *The Social Life of Information*, Harvard Business School Press, Boston, 2000
- [BMM02] Brunetti G., Micelli S., Minora M., *La sfida delle tecnologie di rete: distretti lombardi e veneti a confronto*, FrancoAngeli Milano, 2002
- [But02] Butera F., *Il campanile e la rete - l'economic business e le piccole e medie imprese in Italia* Mondo economico 56, Il Sole24ore, 2002
- [CDR01] Casapulla G., De Cindio F., Ripamonti L.A., *Community Networks and Access for All in the Era of the free Internet: "Discover the Treasure" of the Community*, Keeble L & Loader B D (eds.), “Community Informatics: Shaping Computer-Mediated Social Relations”, Routledge, London, 2001
- [DLR03] De Cindio F., Longhi I., Ripamonti L.A. *The first need is a learning community: Just do it!*, IADIS International Conference e-Society 2003, Lisbon, Portugal, June 3-6 2003
- [Kim00] Kim A.J., *Community building on the Web*, Peachpit Press, 2000
- [LaW91] Lave J., Wenger E., *Situated learning: Legitimate Peripheral Learning*, Cambridge University Press, 1991
- [Pre00] Preece, J. *Online Communities: Designing Usability, Supporting Sociability*, Chichester, UK: John Wiley & Sons, 2000
- [TeG02] Terra J.C., Gordon C., *Realizing the Promise of Corporate Portals: leveraging knowledge for business success*, Butterworth Heinemann, Boston, 2002
- [Wen01] Wenger E., *Supporting communities of practice: a survey of community-oriented technologies*, Draft Version 1.3, March 2001, <http://www.ewenger.com/ewbooks.html>