Piccinini, N., & Scollo, G. (2006). Cooperative Project-based Learning in a Web-based Software Engineering Course. *Educational Technology & Society*, 9 (4), 54-62.

# Cooperative Project-based Learning in a Web-based Software Engineering Course

**Nicola Piccinini**
Università di Verona, Dipartimento di Informatica, Ca' Vignal 2, Strada Le Grazie 15, I-37134 Verona, Italy
piccinini@sci.univr.it
http://www.sci.univr.it/~piccini

**Giuseppe Scollo**
Università di Catania, Dipartimento di Matematica e Informatica, Viale Andrea Doria, 6, I-95125 Catania, Italy
scollo@dmi.unict.it
http://www.dmi.unict.it/~scollo

**ABSTRACT**

Even in self-organized project-based learning, the instructors' role re-mains critical, especially in the initial orientation provided to the students in order to grasp the educational goals and the various roles they may undertake to achieve them. In this paper we survey a few questions proposed to that purpose in a web-based software engineering course, together with relevant answers, we outline the project set-up methodology aimed at providing students with that initial orientation in the laboratory part of the course, we collect a few empirical data out of the latest seven-year history of the course and, finally, we put the presented work in the context of current approaches to software engineering education and draw brief conclusions.

**Keywords**

Problem-based learning, Project-based learning, Cooperative web-based learning, Self-organized learning, Meta-cognitive reinforcement

## Introduction

Since seven years, the Software engineering course offered at the CS department in Verona relies upon the use of a cooperative web-based organization, whereby laboratory project work as well the course organization itself evolve following feedback and proposals made by the students. The educational approach thus matches the profile of "problem-based learning, project-based learning, and collaborative problem solving" Nesbit & Winne (2003). Furthermore, besides using a web-based cooperation platform BSCW (2005) to support interaction, coordination and resource sharing, the web itself is used as a source of knowledge exploration and support to collaborative inquiry learning (Chang et al., 2003; Salovaara, 2005), within a largely self-organized project teamwork.

Nonetheless, the instructors' role remains critical, especially in the initial orientation provided to the students in order to grasp the educational goals settled from the outset, and the various roles and responsibilities they may undertake to achieve those goals. This paper is aimed at corroborating this statement with empirical evidence drawn from the subject course experience. This is presented according to the following organization of the paper.

In the next section, we survey a few questions presented to students in the initial part of the course, to the aforementioned orientation purpose. In the subsequent section, a select blend of answers proposed to those questions are reviewed and discussed, with reference to the main statement of the paper. We then proceed to outline the project set-up methodology that is aimed at providing students with initial orientation in the laboratory part of the course. Empirical data out of the latest seven-year history of the course laboratory are then collected and discussed, again with reference to the main statement. We finally put the presented work in the context of current approaches to software engineering education and draw brief conclusions. A more extensive, albeit earlier (hence less up-to-date) report on this web-based laboratory learning experience is available (in Italian) in Piccinini & Scollo (2005).

## Questions

The following questions are excerpted (and translated from Italian) from the on-line lecture notes, which can be consulted at Scollo (2005).

*Q.1: Use of software engineering principles in web design*

The notes start with an introductory lecture where 1) several analogies are pointed out between software design and web design, and 2) traditional software engineering principles are presented and argued about. The very first group of exercises then looks like as follows.

> We propose a few examples of web design problems, all somehow connected to its rapid evolution character (...) whereas we propose as *exercises*: 1) the identification of software engineering principles, out of those introduced in the lecture notes, that appear relevant to each of the problems proposed, and 2) the development of a rationale for their respective applications to evaluation of solutions to those problems.
> With websites whose management brings frequent changes, users find it useful to get information about website updates, that is, answering the question: What's new? Identify and evaluate at least two different solutions to the problem of how to provide this service.
> etc.

*Q.2: Inquiry into ergonomic qualities of the course website*

In a subsequent lecture, software engineering principles are put in the wider context of quality design rules, some of which are drawn from ancient, philosophical tradition (such as Ockham's Razor, Duns Scott's Consistency Rule, etc.). Use of these rules is then invited to be exercised through inquiry into ergonomic qualities of the course website. To this purpose the following questions are proposed, among others:

> Find aspects (of the structure and/or contents) of the website of this course which violate some (which?) of the proposed quality design rules, and motivate your answer.
> Find aspects (of the structure and/or contents) of the website of this course which satisfy some (which?) of the proposed quality design rules, and motivate your answer.

These questions, as well as those relating to subsequent lectures, share a common premiss, which alludes at expected students' contributions as co-designers of the course website contents:

> We do not propose to consider design qualities of the BSCW system as such, but only qualities that be relevant to the *use* of this system for the educational purposes of this course. The following may thus come into play:
> aspects of the BSCW system as such, that prove relevant to the aforementioned use;
> aspects of its use made by the designers of this workspace (instructors and tutors), who provide a web-based educational service to the course participants, and so are both system users and service providers at the same time;
> aspects of the use of this system and of this workspace made by the users of the afore-mentioned educational service, who, too, contribute to modify the structure and contents of the workspace, hence also have their share of responsibility for the ergonomic quality of the website in question.

*Q.3: Self-referential inquiry into workgroup organization*

As a third, final case of interest we excerpt a couple of questions relating to workgroup organization, that are based on a preliminary working assumption which takes the flavour of a self-referential case study:

> Consider the following problem as a *working assumption*:
>
> > Organize (part of) the documentation that is produced by the participants to this edition of the course, in the form of a website that would be suitable for use by the instructors and participants to future editions of the same course.
>
> With reference to such a working assumption:
> 1. propose working practices meant to help the efficiency of the production process;
> 2. characterize an organizational structure that would prove both adequate to the objectives and compatible with the production context constraints;
> 3. etc.

## Answers and reinforcement

*A.1: Use of software engineering principles in web design*

The first questions listed in the previous section were proposed since the 2002 edition of the course. Following students' feedback, viz. a first series of "systematically wrong" answers, their presentation is now endowed with a supplement of meta-cognitive information, just aimed at clarification and *reinforcement* of the stated goal. Here is what is added to this purpose, and why. The first answers only dealt with the proposed web design problems, with no reference to software engineeering principles whatsoever. An example of such kind of answers is the following:

> The first method to handle such a situation consists in mailing an update report, after every update, where the various changes made are described and documented.
> The second proposal is to display a list of the recent changes as initial page.

This led to inclusion of the following premiss to the exercise:

> The use of software engineering principles, such as those proposed in the lecture notes, proves purposeful to identification as well as evaluation of solutions to web design problems. Each of the proposed exercises asks to validate this statement in a concrete problem. What is thus required is not so much to invent a solution to each of the proposed problems as, rather, to reason about the use of the aforementioned principles to those purposes.

*A.2: Inquiry into ergonomic qualities of the course website*

What makes the following answers interesting, from an educational process viewpoint, is that they reveal a limited inquiry into the proposed subject of investigation. This gives instructors the opportunity to point at further inquiry directions and to interactively refine the students' knowledge and understanding, again with a meta-cognitive reinforcement side-effect. Here are two examples, respectively referring to the two questions quoted in the previous section, where a student's answer is excerpted, followed by the instructor's reply.

> A.: In my opinion an important functionality is lacking within the BSCW system, and this may violate the completeness rule. The functionality which is not present (or I didn't find) in the BSCW is that which would enable one to download a whole folder, including its contents.
> R.: The functionality is available, by means of the creation of a compressed archive of the folder, both for download and for upload.

Note that the student is well aware that s/he may have failed to locate the desired function. The instructor's reply thus works as a solution advice to a student's inquiry problem.

> A.: The very structure of the BSCW system warrants conformance to the orthogonality rule. The organization into folders and subfolders, each of them dealing with a particular subject, allows the separation of distinct aspects and a quicker information retrieval by the user.
> R.: Rather than "warrants" I should like to say "enables", since it's up to the content producers to actually maintain separation of independent aspects by means of their placement into distinct folders.

In this case, the instructor's reply reinforces the alluded students' responsibility as content producers.

*A.3: Self-referential inquiry into workgroup organization*

A seemingly surprising fact about the third group of questions recalled in the previous section is that none of them has ever been answered directly so far. This can be explained as follows. Unlike the previous questions, which are proposed at an early stage of the course, the subject questions occur at an advanced stage, when laboratory project work is well beyond take-off. Answering the proposed questions is an optional assignment in all cases, but with the first, early questions there's no alternative assignment anyway. On the contrary, at the advanced stage when the subject of group organization comes to be dealt with, practice on it may well take place within the self-organized context of each project teamwork.

The proposed questions prove useful nonetheless, in that they inspire similar questions which groups do address within their own self-regulated contribution to the laboratory production work, thereby answering those questions indirectly, viz. with reference to a different, albeit related working assumption. Rather than working as task assignments, thus, questions of this kind act as models for laboratory project set-up and organization, which is the subject of the next section.

The lack of direct answers to the proposed questions, together with their proven usefulness in the development of student projects are thus no failure but rather to be taken as a sign of *success* of a scaffolding strategy, whereby instructors first model the desired working skills to be developed and then gradually shift responsibility to the students.

## Laboratory project set-up

In order to accomplish the laboratory course tasks, students have got to put the concepts and methods learnt from the lectures into practice. This is organized by way of simulation of a (fairly liberal) software production environment, where project ideas and proposals are invited, project teams are dynamically created around them, and team projects are viewed as mildly co-ordinated constituents of a somewhat larger "laboratory project", where industry-like control and co-ordination mechanisms are experimented.

A project idea proposal does not only involve a concise description of the functionality of an intended software product, but is also required to exhibit a clear identification of the pursued educational objectives, mostly specified in the form of a list of intended project deliverables. Here it is understood that the production of a certain deliverable, say a requirements specification, a risk management plan, or a collection of architectural models, is a task that is instrumental to learn how to apply concepts and methodological prescriptions relating to that kind of task—this educational objective is thus designated by the corresponding deliverable.

The intended software product is by no means expected to be actually produced, only the planned deliverables so are. However, in addition to setting educational objectives, a project idea proposal may include quality objectives. These refer to the intended software product, are kept distinct from the educational objectives, and denote the intent to evaluate and document the influence of desired quality characteristics of the specific product on the production process to be planned.

Team projects are largely self-organized, in that all project activities, ranging from project idea formulation and choice of educational objectives through project team composition, production organization and actual execution, are left to the students. Quality of the organization work is a most relevant subject of the final evaluation, which is in principle aimed at evaluating the production process maturity rather than quality of the resulting products.

Despite full-range self-organization, instructors do play a key role of guidance and control, which shows up at least on three subsequent stages of project evolution:

1. Initial explanation, of what we are summarizing in this section, in classroom and by means of on-line documentation, in particular Piccinini (2005).
2. Team project set-up. Before starting each team project work, students are strongly encouraged to seek instructors' advice through a public on-line discussion about their project idea. Other students may of course join the discussion. This starts with a project idea proposal, hopefully including the pursued educational objectives, and sometimes also including a first identification of team members. The instructor may ask for further clarification when any of the following is deemed to hold:
   - the idea is not clearly expressed,
   - the educational objectives are inaccurate, confuse or (as it often happens) missing,
   - the quality objectives, if present, are inaccurate or confuse.

   The discussion continues until no doubt is left and the team is formed.
3. Final exam evaluation. This relates to the whole course program, but draws upon the project work developed in the laboratory.

The second out of the previous three points is the most important one in the laboratory course organization. Since every discussion is publicly held on-line, it gives rise to an interesting case study for course students—those of the current edition as well as those of future ones. Throughout the latest seven-year course history, those discussions have come to build-up a fairly large information corpus that, besides comprising a rich bestiary of potential project ideas (from pure fictionary ones to potential merchandisable products), proves amenable to quantitative analysis, such as that which is presented in the next section.

## Laboratory project evolution, collected data samples

The data sample sizes are displayed in Table 1: a total of almost 1000 students attended the seven course editions.

*Table 1.* Data sample sizes

| edition | students | ideas | projects | % of ideas |
|---|---|---|---|---|
| 1999-2000 | 65 | 12 | 12 | 100% |
| 2000-2001 | 54 | 14 | 13 | 93% |
| 2001-2002 | 167 | 38 | 33 | 87% |
| 2002-2003 | 185 | 45 | 43 | 96% |
| 2003-2004 | 187 | 42 | 38 | 90% |
| 2004-2005 | 143 | 31 | 30 | 97% |
| 2005-2006 | 174 | 41 | 40 | 98% |
| Total | 975 | 223 | 209 | 94% |

*Table 2.* Project idea discussions

| edition | 1999-2000 | 2000-1 | 2001-2 | 2002-3 | 2003-4 | 2004-5 | 2005-6 | Total |
|---|---|---|---|---|---|---|---|---|
| # ideas | 12 | 14 | 38 | 45 | 42 | 31 | 41 | 223 |
| *correct* | 5<br>42% | 5<br>36% | 11<br>29% | 15<br>33% | 21<br>50% | 7<br>23% | 13<br>32% | 77<br>35% |
| *clarified* | 1<br>8% | 2<br>14% | 6<br>16% | 6<br>13% | 4<br>10% | 2<br>6% | 7<br>17% | 28<br>13% |
| *refined* | 0<br>0% | 4<br>29% | 8<br>21% | 8<br>18% | 7<br>17% | 7<br>23% | 13<br>32% | 47<br>21% |
| *edu.oriented* | 0<br>0% | 7<br>50% | 20<br>53% | 24<br>53% | 12<br>29% | 17<br>55% | 19<br>46% | 99<br>44% |
| *qual.oriented* | 0<br>0% | 0<br>0% | 1<br>3% | 4<br>9% | 8<br>19% | 9<br>29% | 6<br>15% | 28<br>13% |
| *team set-up* | 1<br>8% | 2<br>14% | 6<br>16% | 10<br>22% | 11<br>26% | 7<br>23% | 13<br>32% | 50<br>22% |
| *unfinished* | 1<br>8% | 2<br>14% | 8<br>21% | 9<br>20% | 6<br>14% | 8<br>26% | 2<br>5% | 36<br>16% |

The data relating to the current course edition, 2005–2006, should not be considered as final ones, since course exams are on schedule for the rest of the year; students are not required to physically attend lectures, as they may get the course materials as well as organize team projects on-line. They proposed 223 different project ideas; most of these (209, nearly 94%) evolved to a concrete project, run by a working team with well defined objectives.

However, in the majority of the sample cases, the transition from project idea to actual project is not immediate. Table 2 shows, in actual as well as percentual value, the incidence of the most frequent cases of delayed project start-up, as found in the project idea discussions, where corrections or refinements of idea proposals were put forward. This yields a quantitative measure of the relevance of the instructor's role to orientate the project start-up along the desired educational path. The first row in Table 2 lists the number of project idea proposals, briefly “ideas”, per course edition. Subsequent rows are labelled according to the following terminology.

*correct* are the ideas that appear sufficientely clear, accurate and complete since their first formulation. For example, students who débute with:

> Our idea is to design a on-line store of CD's, DVD's and, in general, consumable electronic materials (printer cartdridge, etc.). The customer may consult the on-line catalogue and book the chosen products. There will also be a forum where the customer may comment on the purchased goods.

> The project objectives are:
> - quality control (functionality, reliability, usability, maintainability),
> - feasibility study,
> - risk management.
>
> Group components: [omissis]

All is right and nothing is missing, hence the instructor replies:

> Good job, go ahead!

*clarified* are the correct ideas that receive either additional suggestions from the instructor or requests for further information from the instructor and/or the students.

*refined* are those ideas whose initial description is too vague, inaccurate, or even confuse, for example:

> We intend to design a software for the management of the civil guard emergency units. Units co-ordination will be driven by the emergency type and by the skills and availability of staff members.

Although the use context of the target software is well described, an outline of the desidered functionality is missing, and this makes it difficult to evaluate the idea.

*edu.oriented* are those ideas whose initial description either lacks educational objectives or has such a list thereof, but this is inaccurate or unclear. Here is an example of the first kind:

> The Romantik Tour travel agency has contacted our software house to produce a CD-ROM catalogue that is to be distributed both to customers and to tour operators. Moreover, this catalogue may be later upgraded by the agency.

The lack of educational orientation is pinpointed by the instructor:

> The idea is briefly and well expressed, with respect to functionality. But what about the objectives of your laboratory work?

As an example of the second kind, here is a proposed list of objectives:

> feasibility study, risk management, quality objectives, production process architecture, usability and accessibility, identification of resources and operational constraints, operational plan, time and cost estimation.

The instructor replies:

> The proposal is not clear because objectives and sub-objectives are listed with no logical structure, could you please reorganize them? Moreover, I suggest to add workgroup planning to your objectives, because the group is very large, hence it needs careful management.

The advice is not limited to point out a problem in the formulation of the educational objectives, but it also contains an additonal suggestion aimed at getting the team started with proper attention to effectiveness of its own organization. This example highlights the guidance role of the instructor in operational terms, besides conceptual ones.

*qual.oriented* are ideas that feature quality objectives, but in an unclear manner.

*team set-up* are discussions that, besides clarifying, refining or orienting a project idea, are also exploited to define the project team.

*unfinished* are those idea discussions which leave unanswered questions. For example, the aforementioned discussion of the “RomantikTour” idea belongs to this category, as the students never satisfied the instructor's request.

For each column in Table 2, the sum of the discussion counts placed under the aforementioned categories need not coincide with the total number of ideas, since those categories are not mutually exclusive, but also because a few ideas (approximately 10% of the total number over the seven course editions) directly show up as started projects, without going through a discussion between students and instructors.

After this due premiss, a quick look at the data in Table 2 immediately reveals that the by far most frequent reason for discussion has to do with the educational orientation, whose lack or unclarity affects almost a half of the ideas. Refinement of the functional content of the idea takes the second place, finally followed by orientation with respect to quality objectives. This one featured a rapidly growing trend until the previous course edition, which fact may be explained by the growing importance given by the instructors to quality objectives, in the course of time. The exploitment of project idea discussions as an opportunity to build up the project team exhibits a steadily growing trend throughout the whole sequence of course editions.

Table 3 allows us to evaluate other interesting data about the project start-up discussions, expressed as average values. The first two columns respectively concern the size of the project team and the number of notes posted to those discussions. Because these usually consist of a sequence of questions and answers between instructors and students, we put forward this value as another quantitative indicator of the relevance of the instructors' role.

*Table 3.* Statistical averages

| edition | team size | posted notes | start week | start week *refined* | start week *edu.oriented* | start week *qual.oriented* |
|---|---|---|---|---|---|---|
| 1999-2000 | 4.75 | 1.20 | 5.67 | — | — | — |
| 2000-2001 | 3.45 | 3.43 | 1.86 | 1.50 | 1.43 | — |
| 2001-2002 | 3.43 | 3.92 | 11.37 | 4.50 | 7.50 | 2.25 |
| 2002-2003 | 3.17 | 3.88 | 17.18 | 11.13 | 13.71 | 13.67 |
| 2003-2004 | 3.53 | 4.53 | 11.41 | 17.29 | 9.17 | 10.57 |
| 2004-2005 | 4.07 | 4.32 | 13.60 | 10.14 | 7.41 | 10.65 |
| 2005-2006 | 3.44 | 5.97 | 8.29 | 7.38 | 6.47 | 8.17 |

Starting from the second course edition, for each idea discussion the smallest number of notes is 2, as it happens when the idea is immediately judged to be correctly formulated, whereas more notes are to be expected when problems affect its first formulation. In the first course edition the average value of the number of posted notes falls below 2 because, unlike the subsequent course editions, all project idea proposals were posted to a single forum, where they most often did not receive any feedback from the instructor—that was rather given personally, during project discussion time in the lab. Several other changes have occurred in the course organization over the years, testifying to the fact that this has been a learning experience for the instructors no less than for the students.

The three rightmost columns provide one with a few indicators on how the various start-up discussions, added in the course of time, come to build a sort of collective memory that helps newcomers to correctly formulate their own idea. The term "start week" denotes the number of weeks spent before proposers of a project idea introduce their idea; this number is counted starting from the day when the instructors give a classroom introduction to how should team projects be set up.

The subsequent three columns report the same statistic, but only computed over the project ideas which fall under one of the three aforementioned categories which have been shown to be the main reasons of discussions. Except for one case (precisely *refined* ideas at the fifth course edition), the averages in these three columns are lower than the general average; this fact clearly tells that ideas submitted later are more frequently found correct, probably because proposers have learnt from the experience made by their preceding, eager-to-start colleagues.

It seems that no similar improvement can be ascribed to progress throughout subsequent course editions, for reasons that are as yet to be fully understood. However, there appears to be a significant correlation between the rate of correctness of initial formulation of ideas and average start week throughout different course editions; if $c_y$ denotes the percentual number of *correct* ideas at course edition $y$, while $w_y$ denotes the average start week at the same course edition, our data yield a value of 0.805 for the correlation coefficient between these two statistics.

## Related work

There is no standard methodology for software engineering education. The basic rationale for including student projects in the educational activities is to let students get a foretaste of what a professional in the field is expected to do, and how. Not all software engineering courses include such a laboratory part. Even those which do so, may largely differ in the way projects are set-up and organized, e.g. whether or not do they involve collaborations with industrial sponsors, and, in either case, who is in charge of defining project objectives and task assignments.

Because of the rapid pace of change in software engineering methods and technologies, industrial impact through education is a persistently hot topic in specialized conferences on the subject, see for example the panel CSEET (2006) at the forthcoming edition of CSEET.

The laboratory project set-up in the present case study is motivated by firm reasons of principle. The instructors held a constructivist viewpoint, which may be traced back to Vygotsky's social interaction theory Vygotsky (1978), takes due account of the educational value of training students to take up the responsibility of setting their own learning objectives, and exploits the opportunities of web-based cooperation to implement a scaffolding strategy to that take-up purpose.

While the debate on software engineering education has been most often centered on the specific traits of the discipline so far, the importance of adopting proper educational theories is starting to be recognized—see for example Basili & Basili (2006) at the aforementioned conference. Our approach seems consistent with that direction.

## Conclusions

In this paper we have presented and briefly analyzed a case study in software engineering education, spanning over a seven-year evolution, characterized by a blend of educational techniques: traditional classroom lectures, textbook and lecture notes, as well as a web-based cooperation platform, supporting interaction and self-organization of laboratory projects. The laboratory project set-up in the present case study is motivated not only by reasons of principle, as recalled in the previous section, but also by practical constraints. The largely self-organized character of the laboratory project work turns out to be a necessity, since the bare number of students enrolled in the course would make it impossible to provide a continued guidance and daily organization of all projects by just one or two instructors. In other words, scarcety of human resources puts a severe limit on the level of guidance that could be effectively provided under a different arrangement.

The following conclusions can be drawn from the case study summarized in the previous sections.

First, regardless of their “easiness”, the first questions proposed as exercises always turn out to be the most difficult to get answered properly. The main difficulty is to be traced, in this case, to the novelty of the methodological mindset wherein the initial subject—applicability of software engineering principles to web design case studies—is framed.

Second, the issue of endowing project proposals with clearly stated *educational* objectives has persistently turned out to be the most significant reason for delayed set-up of team projects, despite the provision of detailed guidelines to this purpose in the laboratory edu-cational material. The reason for this difficulty is, again, the novelty of a methodological mindset where metacognition is explicitly meant to play the most significant role. Students are just not used to that, they are rather inclined to focus on the functional features of their intended product than on their own educational gain. This is surely not surprising, at the initial project set-up stage, nor is it necessarily a bad sign; it just highlights the usefulness of scaffolding strategies where the level of guidance provided by instructors is significant at the early stages, and then progressively decreasing.

Third, regardless of the variety of alternative organization models proposed for group work, the vast majority of students privilege the most established one, viz. that which appears to be most popular from the previous editions of the course. This can be seen as an instance of the well-known “popularity is attractive” principle from the theory of network evolution Dorogovtsev & Mendes (2003).

Our final, perhaps most interesting observation is that, even in self-organized project-based learning, the instructors' role remains critical, especially in the initial orientation provided to the students in order to grasp the educational goals and the various roles they may undertake to achieve them. This educational aim seems to be most effectively achieved by prompting and exploiting all available opportunities for meta-cognitive reinforcement, which fosters deeper inquiry capabilities, knowledge and insight.

## References


Basili, V.R., & Basili, P.A. (2006). *Software Engineering Instruction and Education Theory: a Dialogue*, retrieved July 24, 2006 from http://db-itm.cba.hawaii.edu/cseet2006/Basili.php.

BSCW (2005). *Basic Support for Cooperative Work*, retrieved October 2, 2006 from http://bscw.fit.fraunhofer.de/.

CSEET (2006). *CSEET 2006 Panel: Industrial Impact through Education – Lessons Learned from Barry Boehm's Contributions to Software Engineering*, retrieved July 24, 2006 from http://www.sbl.tkk.fi/CSEET06panel.

Chang, K.-E., Sung, Y.-T., & Lee, C.-L. (2003). Web-based collaborative inquiry learning. *Journal of Computer Assisted Learning, 19* (1), 56-69.

Dorogovtsev, S., & Mendes, J. (2003). *Evolution of Networks*, Oxford, England: OUP.

Nesbit, J., & Winne, P. (2003). Self-regulated inquiry with networked resources. *Canadian Journal of Learning and Technology, 29* (3), retrieved October 2, 2006 from http://www.cjlt.ca/content/vol29.3/cjlt29-3_art5.html.

Piccinini, N. (2005). *Norme d'uso del BSCW. Integrazione per il lavoro di laboratorio*, retrieved October 27, 2006 from http://amarena.sci.univr.it/edu/pub/bscw.cgi/2482086/NormeUsoBSCWLab-in.html.

Piccinini, N., & Scollo, G. (2005). Fare laboratorio in rete: tracce di un'esperienza di web-learning. *DIPAV Quaderni, 12/13*, 143-164.

Salovaara, H. (2005). An exploration of students' strategy use in inquiry-based computer-supported collaborative learning. *Journal of Computer Assisted Learning, 21* (1), 39-52.

Scollo, G. (2005). *Architetture software e Ingegneria del software, Note delle Lezioni*, retrieved October 9, 2006 from http://amarena.sci.univr.it/edu/pub/bscw.cgi/2481306/AIS0506-in.

Vygotsky, L. (1978). *Mind in Society*, Cambridge, MA: Harvard University Press.