# Tools for Supporting Hybrid Learning Strategies in Open Source Software Environments

Francesco Di Cerbo[1], Paola Forcheri[2], Gabriella Dodero[1], and Giancarlo Succi[1]

[1] Center for Applied Software Engineering
Faculty of Computer Science
Free University of Bolzano-Bozen
Piazza Domenicani 1
39100 Bolzano-Bozen, Italy
{fdicerbo,gdodero,gsucci}@unibz.it

[2] Institute for Applied Mathematics and Information Technologies
Italian National Research Council
Department of Genoa, Italy
Via De Marini, 6
16149 Genoa, Italy
forcheri@ge.imati.cnr.it

**Abstract.** In this paper, we illustrate how a cooperative learning paradigm may benefit from cutting edge e-learning techniques. We use Web 2.0 resources (especially AJAX) to fulfill requirements for an interactive-constructivistic "learning space", extending an existing Free/Open Source Software Learning Management System, to create a cooperative and community-based learning space adherent to our proposal. The paper shows also how to use our toolset on two case studies.

**Keywords:** Open Source Software, virtual learning space, social communities

## 1 Introduction

The development of the so-called "soft-skills" has become an indispensable complement to any discipline, be it technology oriented or in the humanities. With "soft skills", the European Community [1], identifies abilities in

- presentation of knowledge
- effective web search
- project work
- team work
- problem solving

These skills can be considered of paramount importance in any curricula, and as they can be effectively improved through the use of technologies, it is possible to refer to them as ICT-enhanced skills.

Methodologies have already been identified, which may be applied in order to achieve such results. Examples are "problem-based learning" [2], "troubleshooting"

[3] or “case-based teaching” [4]. Several pilot experiences already support this claim (see for example [5] and [6]), even if an exhaustive experimentation involving e-learning specialists and students at various levels and disciplines is not yet completed.

As it is well known (see [7]) these methodologies, inspired to a constructivist basis ([8]), view learning as the result of student-centered activities, addressed by intentions and reflections, requiring to carry out authentic tasks and proposing concrete goals. Moreover, they integrate individual and collaborative work, thus including a socio-constructivist component ([9]).

Starting from the experiences collected in the European Project I*Teach ([10]), we started rethinking about the tools we used to effectively support our teaching experiences, and decided to initiate a new effort to put in practice the lessons we learned.

The work we present consists in the development of a technological infrastructure, suitable to support active learning methodologies. This development mostly concentrated on the possibility to pass from the learning environment concept (the status-quo of the market) to that of “dynamic learning space” (DLS, similar to [10]) which is the novelty of our approach. In such a DLS, people involved in learning are identified by their roles, attributes and behaviors, and meet each other to collaborate to the negotiation and construction of knowledge.

The concept of “dynamic learning space” is not originating in the virtual environment by itself. Indeed, in a real (not virtual) learning space, both teacher and his/her class participate to the creation of a shared knowledge; they build up meanings and concepts where every individual has its own role inside the process. Teachers have the supervision of the community, steering global effort towards learning targets, and every student may / must contribute to the global learning of the community.

In case of a virtual teaching, such as in hybrid and e-learning scenarios, a teacher should rely on a set of software tools, letting him involve his/her class, like during lessons in presence. We developed software to help teachers to coordinate such virtual communities, specifically at interaction level, in order to form a social network. Such a software has been called DIEL, Dynamic Interactive E-Learning system.

By DIEL, co-construction of learning materials and concepts is achieved through virtual experiences. Since it gives a precise and “automatic” track of activities conducted, without the distractive effect caused by recording them by hand, students and teachers may concentrate on the learning process. A detailed description of the pedagogical aspects taken into account in our design is available in [11].

The paper includes a description of the approach we followed, its implementation as well as two case studies to show the possibilities of DIEL.

## 2 The approach

Our approach in the design of DIEL aimed at supporting learning activities using ICT technology. We targeted learning communities at academic level, but due to the easiness of use and the simple metaphors we chose, the tool may be used also for students of lower ages.

To implement DIEL, we decided to extend an existing Learning Management System (LMS), reusing a part of its tools and services, that are still useful in our vision. The Moodle open source LMS [12] was our starting point. Moodle is an open source environment which is broadly used around the world (with over 36,000 registered sites with 14 million users in 1.4 million courses, according to the Moodle website) and supports more than 70 national languages. Its adoption level leads us to choose Moodle as our target platform. The actual implementation at each installation may vary due to national options, and portal contents, visibility policy and interaction functionalities are implemented by such system as custom installation features, in order to better match actual needs of the users' community.

What really represents a discontinuity with all established solutions is a new metaphor, designed to highlight constructivistic approach in contents creation. Exploiting new technology capabilities (Java applets and AJAX, the engine of Web 2.0 paradigm, see [13]), DIEL creates a virtual environment where interactions are welcomed and eased, and where every community service contributes to the creation of a common knowledge as part of a structured learning process.

Each involved person, students as well as teachers, is free to operate and move in our virtual classroom: it is a place where to put opinions or contents, to meet the classmates, even to find amusement, without a fixed interaction stereotype. In such virtual environment, everyone is free to find his way to learn, in conjunction with the others and under teachers directions. Previous research in Human-Computer Interaction (for instance, the work of Gräther and Prinz [14]) on community and presence awareness, and especially the concept of social translucence [15] has been used as the basis for our metaphor. This concept allows each user of our community to be aware of what other users are doing in every moment.

Every user is associated to an avatar, which is free to move in a web page, where logical proximity of activities is naturally mapped into physical proximity of the avatars in the virtual space. In Figure 1, it is possible to see an example of an interaction between an avatar's client and a course material, for instance a book available to students. The avatar moves onto the picture representing the material, and a pop-up menu will appear, asking for possible actions to perform (in the example, the alternatives are: open, download, and cancel current action). Similar behaviors are associated to interactions between avatars and other elements (for instance, rooms, selected regions in virtual spaces connected through special passages rendered as doors), and also among avatars and avatars, such as the opening of a private chat session.

Other features that can be used in the virtual space include those which are usually supported by LMSs, like forums and wikis. We added to this basic set, a videoconferencing system based only on Java (applets and services) in a client/server architecture, and an AJAX [16] whiteboard, shared among the community. Both are browser independent and do not require additional software installation at client side.

DIEL is designed to be used as an additional set of feature to sum up to the original Moodle ones. In fact, DIEL supports group interactions even if conducted at unusual timings, letting the users have a coordination point where to meet, after agreeing a specific appointment or even completely by chance. The latter possibility is permitted by the immediate feeling given by seeing the avatars online and the respective actions that are performed. DIEL can be also used to configure a part of a course created in

Moodle; in fact, through the AJAX visualization, it is possible to create rooms as well as resources, besides the DIEL activities. This feature can be a help to speedup the Moodle learning curve for teachers that are not particularly skilled in the use of ICT.

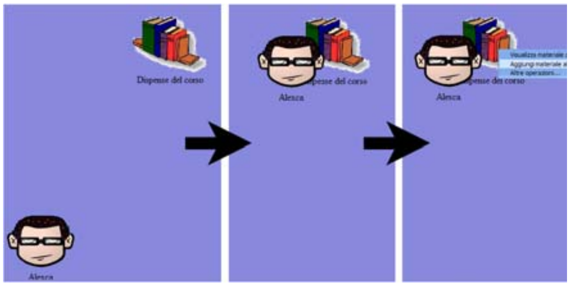

**Fig. 1.** Example of an interaction between an avatar and course material

## 3 Implementation

Our software consists on two main subsystems: one implements all the operations at client side, while the other is responsible for managing all the interactions between users and the system. We defined  a set of system behaviors, called actions, regarding possible interactions between clients and server. Then, we also defined a set of widgets and utility objects to be used at client-side. Each object is associated to a number of actions that are needed to perform its own tasks. All these definitions have been formalized in XML documents, in order to be independent from any programming language (neither client- nor server-side, see example in Table 1). We refer to such definitions as “XML Interfaces”.

**Table 1.** Example of a DIEL action using XML

```
<?xml ... ?>
<action name="move">
 <param name="avatar_name"/ value=”Alexca”>
 <param name="start_xy"/ value=”10,15”>
 <param name="end_xy"/ value=”20,30”>
  <url>...</url>
</action>
```

The development of objects both at client and at server side was based on such XML interfaces. Objects may exchange messages in XML format, if they can parse XML interfaces to produce XML requests accordingly. A representation of the two subsystems as well as XML interfaces is available in Fig. 1.

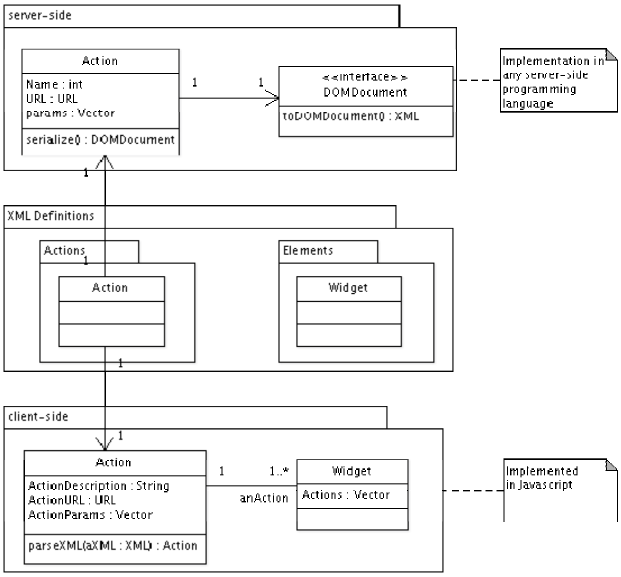

**Fig. 2.** Diagram of subsystems

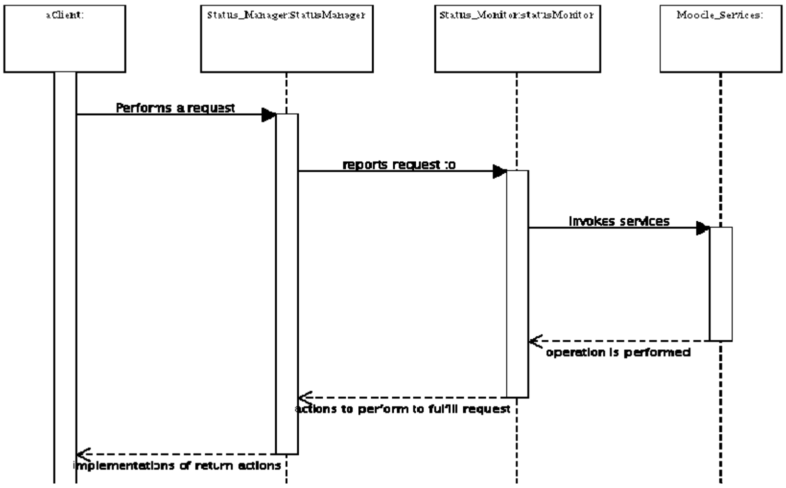
aClient:
Status_Manager:StatusManager
Status_Monitor:statusMonitor
Moodle_Services:
Performs a request
reports request to
Invokes services
operation is performed
actions to perform to fulfill request
implementations of return actions

**Fig. 3.** Sequence diagram of interactions between Status Monitor and Status Manager

Two special entities are responsible for communication between client and server: Status Manager and Status Monitor. Both of them embed the concept of “status”; as DIEL is focused on live social interactions, we require that all the users share the same experience. As a consequence all users also have to share the very same status; we implemented this requirement keeping the status of the whole application at server-side, and exchanging periodic updates with clients. Once an action has been performed by one of the clients, all the others are notified within 1-2 seconds. This mechanism is based on the mentioned Status Monitor and Status Manager. A sequence diagram showing their interactions is available in Figure 3.

Status Monitor is an object managing all the client side tasks and operations. There are as many instances of Status Monitor as there are clients. In fact, it is responsible for creating the initial environment (avatars, rooms, widgets and so on), and to track any modification to its internal status (i.e., movement of the user's avatar, insertion of a new message in the real-time chat and so on). Once a modification happens, Status Monitor sends an update message to its server-side counterpart, Status Manager, receiving back (in the next update message) the actions to perform to show the actual status of the application to the user.

Status Manager sends to all Status Monitors an update message at predefined intervals; this message is created considering the past history of the Status Monitors since the last update message. The system has one and only Status Manager, as we need a central coordination point for the whole system.

## 4 Case studies

We describe learning activities proposed within the project I*Teach [5], and their representation using DIEL. The aim of this section is to show concrete examples on what can be done with DIEL. The two cases show the flexibility of our metaphors, which allow different strategies with the same tool. The cases studies are held in a hybrid learning environment, including classroom face-to face lessons, group interactions and discussions, and online activities. The first one is more oriented to promote discussions and co-construction of contents, while the second shows how heterogeneous groups can find a support for their respective activities.

### 4.1 Case 1: Budget

**Description:** an activity focused on spreadsheet, supervised by the teacher, aimed at introducing spreadsheets concepts and tools. The activity is centered on the following problem: “A young family needs to know the flow of expenses in order to see if it is possible to limit them. Thus, it decides to build a household budget”. Build such a budget and comment on it.

**Active learning method(s):** problem based learning

**Learning objectives:**

- introduce and analyze the budget concept
- Improve basic knowledge about worksheets and about (basic) functionalities of a specific spreadsheet program

**Process:**

- Task 1: Introduction of the budget problem as a decision problem
- Task 2: First planning of the budget
- Task 3: Analysis of some functionalities of a spreadsheet program and development of the budget
- Task 4: To develop the report on the work done

- Task 5: Comparison of the result with the expected outcomes

**Case 1 using DIEL.** DIEL virtual environment is partitioned into a set of rooms to be visited in sequence, as can be viewed in Fig. 4**.**

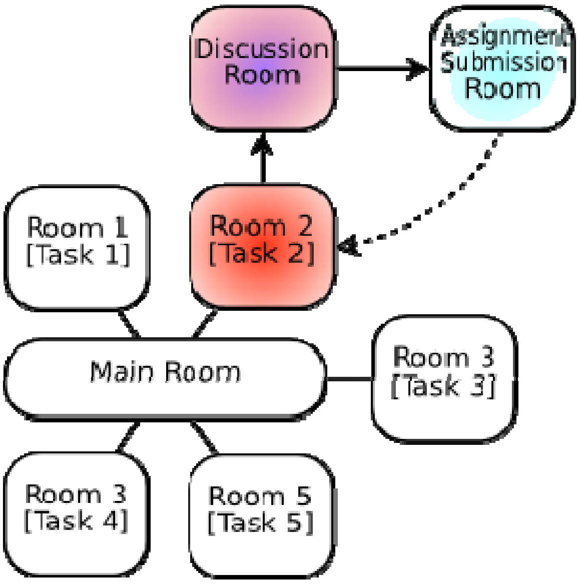

**Fig. 4.** An implementation of I*Teach Case Study 1 using DIEL

Initially, the teacher configures the system in order to enable the students to access the Main Room only. In that room, the teacher places a part of the learning material, as well as other resources that have to be available during all the duration of the course. Students gain access to the different rooms as the course develops, and the teacher decides when and what to show or to close, according to the progress of the course. Fig. 4 shows an example of the implementation of Task 2: such activity leads to the creation of an assignment, to be developed in groups, and usually a group discussion is required to clarify how to realize and structure the final budget. So, students, through their avatars, meet and exchange opinions, achieving a co-construction of shared knowledge, using audio/video services and/or a whiteboard, in a room (the discussion room) contiguous to the one of Task 2. When finished, they move and submit their work in another linked room (the assignment room), where they find a resource that is mapped on the existing assignment service exposed by Moodle. Finally, the teacher closes the discussion room, letting students to access Task 2 room, to fetch the didactic material eventually stored there. This way, DIEL rooms are a convenient metaphor to promote community interactions, through a more involving visualization and an integration of a set of useful community tools.

### 4.1 Case 2: web site creation

**Description:** a group activity project to create a web site against the practice of doping into sport activities. Involved students have different backgrounds; in fact, some of them are studying biology, others computer science. Each student will develop those activities which are best suited for her/his study plan.
**Active learning method(s):** project based learning
**Learning objectives:**

- Analysis skills in understanding results coming from a web search
- Skills to produce a web text and a web biography
- Principles of web design

**Process:**

- Task 1: After dividing the class into heterogeneous groups, every group chooses a specific topic to explore (the risks to the health of doping, the importance of fair play and others)
- Task 2: The groups collect a bibliography on the argument, organize it and produce a wrap-up document
- Task 3: After a discussion of the material collected held at class level, every group updates its documents and transforms them into web pages, eventually adding images and videos
- Task 4: The last phase is the publication of the web pages created into a site for the whole project

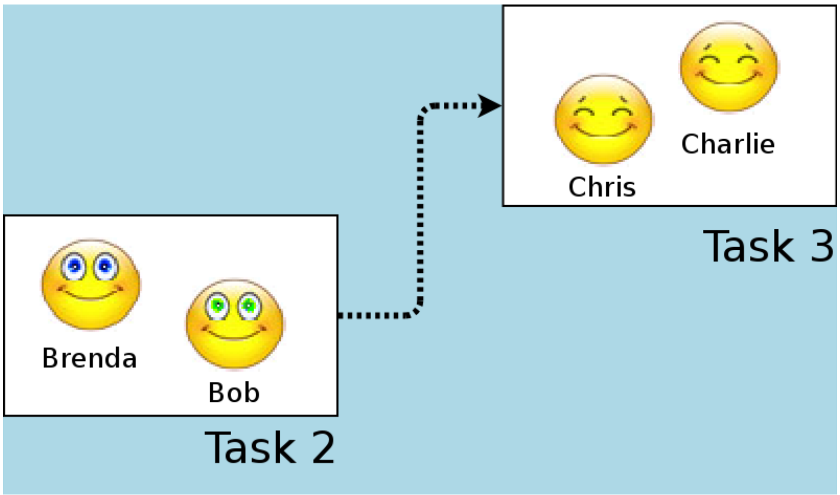

**Fig. 5.** A step in I*Teach Case Study 2 using DIEL

**Case 2 using DIEL.** Case 2 can be implemented as a set of rooms, much in the same way as in the previous example. Fig. 5 shows a possible status of the virtual learning space after Task 1 is finished. The group has agreed with the teacher that Task 2 is accomplished by two biologists, Bob and Brenda, who now are in Task 2 room. Two other group members, the computer scientists Chris and Charlie, in Task 3 room, are waiting for Task 2 to be completed. They will take over with web page implementation as soon as the biologists will "bring" them the collected files, images etc.

## 5 Conclusions

We have described the pedagogical approach, and implementation technologies that we have been employing in order to support social networking inside a new generation LMS, DIEL. Case studies support the claim  that such new interaction models may encourage the teachers to use active learning methodologies in the virtual (as well as in the real) learning environment.

DIEL extends the well-known Moodle Free/Open Source software with a “dynamic learning space”, providing a novel interaction model. At present, the system has been deployed on Moodle 1.8.3, and we are starting to extensively experience it with users (university teachers and students).

To evaluate our solution, we plan to collect both qualitative (through user assessment) and quantitative information. The latter come from automatic tools for collecting process metrics [17] already available and developed at the Free University of Bolzano-Bozen [18]. This will allow us to discover usability patterns inside users' experience, providing a base of evidence useful on two different scenarios: a technical one, to analyze introduced improvements and eventually refine them, and the pedagogical one, to  bring quantitative confirmations  for qualitative considerations.

## 6 References

1. Progress reports of the European Commission’s working groups Improving education of teachers and trainers and ICT in education and training (2003).
2. Ward, J.D. and Lee, C.L., "A Review of Problem-based Learning",  Journal of Family and Consumer Sciences Education, vol. 20, n. 1, pp. 16-26 (2002)
3. Jonassen, D.H., Hung, W., "Learning to Troubleshoot: A New Theory-Based Design Architecture", Educational Psychology Review, vol. 18, n. 1,  Springer , 233 Spring Street, New York, NY USA , (2006)
4. Jonassen, D.H., Hernandez-Serrano, J., "Case-based reasoning and instructional design: Using stories to support problem solving", Educational Technology Research and Development, vol. 50, n. 2,  pp.  65-77, Springer Boston, (2002)
5. Stefanova E., Sendova E., Nikolova I., Nikolova N., “When I*Teach means I*Learn: developing and implementing an innovative methodology for building ICT-enhanced skills”, Proceedings of the IFIP Conference Informatics, Mathematics, and ICT: a ‘golden triangle’ IMICT 2007, Northeastern University, Boston, MA,  (2007)
6. Dodero G., Ratcheva D., Stefanova E., Miranowicz M., Vertan C., Musankoviene V., The virtual training center: a support tool for teachers community, Proceedings of Balkan Conference in Informatics (BCI 2007), Demetra Ltd, (2007)
7. Barak, M, “Instructional principles for fostering learning with ICT: teachers' perspectives as learners and instructors”, Education and Information Technologies Volume: 11, Issue: 2, pp. 121-135 (2006)

8. Bruner, J., Toward a Theory of Instruction. Harvard University Press, Cambridge,MA (1966)
9. Vygotsky, L.S., Mind in Society, Harvard University Press, Cambridge,MA (1978)
10. Pfister, H.-R., Schuckmann, C., Beck-Wilson, J., & Wessner, M. The metaphor of virtual rooms in the cooperative learning environment CLear. In N. A. Streitz, S. Konomi und H. Burkhardt (Eds.), Cooperative Buildings - Integrating Information, Organization, and Architecture. Lecture Notes in Computer Science, Vol. 1370 (pp. 107-113). Springer, Berlin (1998)
11. Di Cerbo F., Succi, G, "A proposal for interactive-constructivistic teaching methods supported by Web 2.0 technologies and environments," Database and Expert Systems Applications, 2007. DEXA '07. 18th International Conference on , pp.648-652 (2007)
12. Dougiamas, M., Taylor, P.C., “Moodle: Using Learning Communities to Create an Open Source Course Management System”, Proceedings of the EDMEDIA 2003 Conference, (2003)
13. O'Reilly, T., “What Is Web 2.0”, accessed on 12 Dec 2005, at http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
14. Gräther, W. and Prinz, W., “Supporting cooperation awareness in common information spaces”, in Supporting the Social Side of Large ScaleSoftware Development, co-located with CSCW 2006, available at http://research.microsoft.com/hip/papers/ssslssdProceedings.pdf
15. Erickson, T. and Kellogg, W.A., "Social translucence: An approach to designing systemsthat mesh with social processes.", Trans. Computer-Human Interaction, vol. 1, n. 7, ACM Press (2002)
16. Murugesan S., "Understanding Web 2.0", IT Professional, vol. 9 , n. 4, pp. 34-41, IEEE Educational Activities Department, Piscataway, NJ, USA (2007)
17. Humphrey, W., Introduction to the Personal Software Process, Addison-Wesley (1997)
18. Scotto M.,Vernazza T.,Sillitti A., Succi G., “Managing Web-Based Information”, Prroceedings of ICEIS Conference, pg. 575-578, (2004)