

Exploring the Potentials of Networked-computing Support for Face-to-face Collaborative Learning



**October 1, 2006
Crete, Greece**

Organizing Committee

Wouter van Diggelen (Utrecht University, The Netherlands)

Vittorio Scarano (University of Salerno, Italy)

Program Committee

Shaaron Ainsworth (UK)

Jerry Andriessen (The Netherlands)

Michael Baker (France)

Annie Corbel (France)

Reuma de Groot (Israel)

Joerg Haake (Germany)

Beatrice Ligorio (Italy)

Paul P. Maglio (USA)

Jakita N. Owensby (USA)

Introduction: Exploring the Potentials of Networked-computing Support for Face-to-face Collaborative Learning

Wouter van Diggelen¹ and Vittorio Scarano²

¹Utrecht University, Research Centre Learning in Interaction,
Heidelberglaan1, 3584 CS Utrecht, The Netherlands.

²ISISLab

Dipartimento di Informatica ed Applicazioni “R.M. Capocelli”,
Universita' di Salerno, 84081 Baronissi (Salerno), Italy.
{w.vandiggelen@fss.uu.nl / vitsca@dia.unisa.it}

Much research into technology-enhanced learning reflects a future of online collaboration, distance learning and virtual teaching¹. These visionary views consider networked-computing support primary as a means to bridge time and space. It is assumed that these collaborative technologies connect learners who couldn't collaborate otherwise. This partial orientation towards networked computing may limit our understanding of the potentials of such technologies for collaboration and learning. Collaborative technologies may also provide effective support for learners who meet face-to-face to collaborate, to discuss and to solve problems [2].

Scenarios studies indicate the school will remain the most important place for learning. Most learning in the nearby future still takes place on-campus, although 'blended mode learning' with a strong ICT component is widely used. ICT use has become commonplace, but it has not radically affected the nature of the teaching and learning. It gradually reshapes traditional on-campus practice [1].

1 Computer support for face-to-face learning situations

The starting-point of the workshop are the notions that *face-to-face learning situations* are an important setting for collaboration and learning *and* that collaborative technologies can support these processes effectively. We believe that one of the most important challenges for technology-enhanced learning is to provide learners who are in same room with the appropriate technologies that will facilitate their collaborative learning activities. The aim of the workshop is to set some directions of how this may be achieved.

We make a distinction between three situations of *technology-enhanced collaborative learning* (Fig. 1).

¹ For example, the vast majority of research on networked-learning environments presented on the international conference on Computer Supported Collaborative Learning (CSCL2003) focused on on-line, virtual meetings between learners [3].

In the *first situation* (upper right corner of figure 1), learners interact with a standalone computer model that represents and simulates a certain problem situation. A computer model typically displays processes that change with respect to time. Learners can manipulate the model and get feedback about their intervention by running a simulation. This form of technology-enhanced learning – sometimes referred to as ‘*single-display groupware*’ (SDG) – has received some investment in terms of research.

The *third situation* (lower right corner of figure 1) represents networked learning environments that aim to *connect learners who are dispersed in time and/or space*. The majority of research into technology-enhanced learning focuses on this type of collaboration. For many researchers, it represents the archetypal context for computer supported collaborative learning (CSCL).

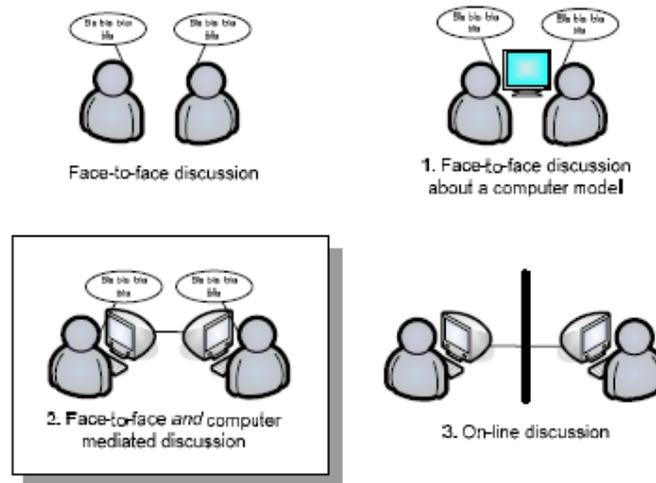


Fig. 1. Three technology-enhanced collaborative learning situations.

The participants of the workshop will focus on the second situation (lower left corner of figure 1): networked-computing support for face-to-face collaborative learning situations.

2 Workshop Theme

The theme of networked-computing support for face-to-face collaborative learning has received relatively little attention within the educational community. Still, it seems to be a promising direction for technology-enhanced learning. Collaborative technologies have the potential to create a sustainable effect on classroom practice. The main objective of the workshop is:

to examine face-to-face collaborative learning situations and to discuss the potentials of networked-computing support for these situations.

The workshop theme will be approached from three perspectives: pedagogical, technical and the perspective of the researchers. These three perspectives are addressed in three different sections.

2.1 Pedagogical perspective

Section I focuses on the pedagogical aspects. Overdijk and van Diggelen focus on the way student groups interact with educational technology. They state that educational technology isn't a stable factor but gets its meaning in practice when students work with the technology. They refer to this process of adaptation as technology appropriation. Technology appropriation helps us to understand why and when new educational technologies work in practice.

Tateo et al. stresses the importance of Participatory Design (PD) to get a better fit between new educational technology and user's activities. PD could reveal issues that may hamper the introduction of educational by involving those who are most affected by the design, i.e. the teachers. Teteo et al. present an explorative study that they carried out among a group of Italian teachers. They conclude that pedagogical support is needed for successful implementation of educational technology in the classroom.

Lotan-Kochan et al. also focus on the teacher. They found that teachers identified several tasks that are crucial during computer supported collaborative learning. The challenge is to provide teachers with the appropriate support – technological as well as pedagogical – that enables them to carry out these tasks effectively.

2.2 Technical perspective

Section II focuses on technological aspects of computing support for face-to-face collaborative learning situations.

Malandrino and Manno present a computer-networking architecture that takes into account the specific requirements that arise from the on-campus technical situation. They present an architecture whose goal is to minimize impact on management and leverages on the LAN setting to ensure extendibility, easy deployment and a uniform work environment by hiding the client-server architecture with a dynamic discovery protocol for bootstrap.

De Chiara and Volpe discuss the development of FireFly, a modular system that allows extensibility and composability. FireFly is written using AJAX, a set of technologies for developing rich web-based application that follow the client-server paradigm. Authors' objective is to develop a client-server system that can be executed smoothly on usual desktop PC, requesting the lesser possible installation effort and achieving enough expandability to allow further extensions'.

2.3 Researchers perspective

From the articles that take a pedagogical perspective one can conclude that there is still a lot of research needed that will increase our understanding of when, how and

why new educational technologies work in practice. Researchers could benefit from tools that automate the collection, transcription and analysis of face-to-face and computer-mediated actions and interactions. *Corbel, Girandot and Lund* addresses this topic in their article. They ‘propose a model of designation and extraction of parts of human interaction corpora using the anchor and link concepts that allow for experimenting on the reuse of analyses of human interactions’.

References

1. Collis, B. and van der Wende, M. (2002). Models of technology and change in higher education: An international comparative survey on the current and future use of ICT in higher education. Center for Higher Education Policy Studies (CHEPS), Enschede, The Netherlands.
2. Tyran, C.K. and Shepherd, M. (2001). Collaborative technologies in the classroom: A review of the GSS research and a research framework. *Information Technology and Management*, 2, 395-418.
3. Wasson, B., Ludvigsen, S. and Hoppe, U. (2003), Designing for change in networked learning environments. Proceedings of the International conference on Computer Support for Collaborative Learning (CSCL 2003), Taipei, Taiwan.

FireFly: Lightweight AJAX System for F2F-CL

Rosario De Chiara, Marco Volpe

ISISLab - Dipartimento di Informatica ed Applicazioni
Università di Salerno,
84084, Fisciano (SA) – Italy
E-mail: dechiara@dia.unisa.it, volpemar@gmail.com

Abstract. In this paper is illustrated the development of *FireFly*, a system for the *Face to Face Collaborative Learning* (F2F-CL). *FireFly* is written using AJAX, a set of technologies that allows to create rich web-based application. The challenge is to develop a system able to be executed on usual desktop PC, requesting the lesser possible installation effort and achieving enough expandability to allow further extensions. The system is currently under development, but the prototypes and the early tests confirms that the use of AJAX technology for this kind of application is suitable and deserves further investigations.

1 Introduction

In this paper we report the results of our experiences in implementing a system for F2F-CL, *FireFly*, exploiting the set of technologies known under the name of AJAX. In our intentions the system must provide the following features:

- Easiness both in the installation and management: virtually *zero* installation effort. This is particularly critical considering that installations are usually made on tens of personal computers.
- Low-end hardware: this can be an issue in various scenarios, for instance schools.
- No Internet connection required: Internet connection is not available everywhere and even where it is available it can be subject to heavy controls and limitations.
- Need for a cross platform solution: platform independence is appreciated from the point of view of the developer, the same application for every machine, and from the point of view of the user that has not to face the problem of different *User Interfaces* (UIs).

The intersection of all these limitations excludes totally or partially some traditional solutions, for various reasons:

- Java applet downloaded from a server located somewhere in the world: this solution would allow to avoid to face legacy hardware (at least for the server) allowing to provide high level services through traditional Java applet software. This kind of solution rely on a trustworthy Internet connection that is not always available. On the other hand Java applications are, of course, cross platform.
- Local Java applications: this solution is suitable for installations on machine not connected to the Internet, but can meet the limitations of computers on which the software cannot be installed. The “install nothing” policy is often enforced by system administrators as a radical solution to viruses and malware.
- Native language solution: a native language solution is not cross-platform by definition. As a pro it can deeply exploit the hardware.

Starting from all these considerations we have pondered about the use of AJAX technology would present some interesting points on its favor.

2 AJAX

The term AJAX is the contracted form for the expression *Asynchronous Javascript And XML* [13]. AJAX is a set of technologies at the hearth of which there is the capability of modern browsers (Mozilla Firefox, Microsoft Internet Explorer, Apple Safari, Opera) of managing an API (*Application Program Interface*) called `XMLHttpRequest`. This API, available within browser through Javascript, allows to transfer data to and from a web server using HTTP. This data transfer is carried out over an independent connection channel and the moved data are formatted in XML.

`XMLHttpRequest` is particularly important because it allows *asynchronous transfers* of data between the client and the server, and this permits to break the constrain of using the traditional form submission mechanism used in HTML [6]. Using `XMLHttpRequest` is just one of the elements that made up the AJAX technology, after the data are moved asynchronously between client and server the next step is to update the user interface in order to reflect the results of this data exchange. The user interface is managed through an HTML web page that is in charge for displaying data and gathering user inputs. Being the data updated asynchronously the user interface must be update in the same manner without a page reload. To obtain this, two well known ways of designing web pages are used: XHTML and CSS, in order to define the styles of the various components (text, labels, buttons etc. . .), and DOM (*Document Object Model*) to address the components of the page that are intended to be modified.

What is really new in AJAX is not the set of techniques but the way this techniques are used to meet a goal that is to use the browser and the network as *a platform for implementing interactive web applications*. The combined effort of XMLHttpRequest for exchanging data and the dynamic look and feel provided to the web pages by the use of XHTML and DOM enable the developer to create applications like GMail [5], Writely [10] and YouOS [12]: GMail is a web mail application, Writely is a cooperatively usable word processing and YouOS tries to mimic the basic behaviors of an operating system (actually a window manager).

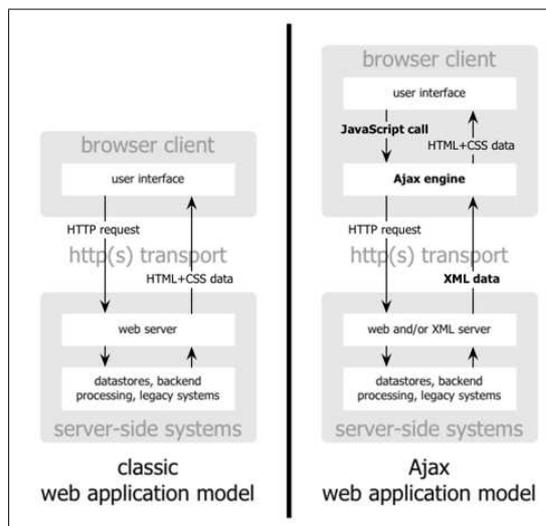


Fig. 1. Comparison between the classic web application and the AJAX web application model. (Image from [13])

In Figure 1 is compared the classic web application model and the AJAX model, on the left is shown the traditional form-driven application, while on the right there is the AJAX model in which the AJAX engine presence within the browser is emphasized. The data exchange between the client and the server is XML and this clearly request a server able to parse and create well-formed XML to send it back to the client.

3 System description

The system is implemented through a suitable configured web server that will provide the application to the clients through an HTML page. A client

for *FireFly* is any modern web browser with not particular settings at all. To let an ordinary PC act as a server for some tens of clients we have focused our attention on finding a computational light web server that would not request too much computational effort.

3.1 Functionalities

The system architecture is quite simple, the software is installed together with the web server. The server part is written in PHP while the client part is Javascript, being it executed in a browser. To boot the system it is enough to execute the web server that simply will be waiting for connections. The participants can enter the system just pointing their browser to the server IP address. An authentication screen will be presented to log into the system.

The system currently provides two tools for the collaboration. One tool is a traditional unstructured chat in which contribution from users are just appended. The other tool is a threaded chat in which the contributions can be structured in a hierarchical manner.

Another task carried out by clients is the gathering of all input, this is performed by Javascript functions in execution within the browser.

Worth nothing is the fact that the system does not use any database engine, everything is stored in XML files and in order to avoid wasteful parsing of huge XML files containing more days of interactions, files are timestamped and rotated everyday in order to keep their size reasonable. These XML files could be used as input for trace analysis softwares a limitation of these traces is that they are coarse grained because of the architecture of *FireFly*.

3.2 Lightweight web servers

Clients interactions will be managed from the web server through CGI (*Common Gateway Interface*) scripts written in PHP. The ability of running CGI scripts is the sole feature a web server has to provide to host AJAX applications. We have compared three solutions, choosing among light web servers: Sambar Server [8], lighttpd [7] and ghttpd [4].

Sambar Server Sambar server is a framework that provides a wide range of different servers (DHCP, SMTP, FTP etc...). The purpose of Sambar is to allow with just one choice, to set up a complete set of services. Sambar is available for both Linux and Windows and is fully configurable through a web interface. A stripped down version of the server is provided for free and it is closed-source.

lighttpd lighttpd is small footprint web server. It is Open Source licensed under the revised BSD license. It is designed keeping in mind the memory and CPU occupation, no matter this it provides a complete set of feature that allows it to be compared with Apache [1]. lighttpd also support FastCGI [3] that is an extension to CGI designed to provide high performance without the limitations of server-dependent solutions. lighttpd is the server currently used for the development of *FireFly*.

ghttpd ghttpd is a small web server released under the GPL. It provides CGI but not FastCGI. It is available just for Linux and Unix.

FireFly is currently using lighttpd for various reasons: operating system independent solution, simple configuration, availability of FastCGI etc. . . . Being *FireFly* an AJAX application it is actually *web server independent*, because its logic is just a collection of standard HTML files and PHP sources that can be installed in whatever CGI-aware web server available.

3.3 User interface issues

The use of AJAX often raises critics about the usability level perceived by users, mainly because the UI has to be implemented using ad-hoc Javascript libraries (see [11], [9], [2]). The UIs rendered using these libraries can be non standard from a user point of view, so a particular attention must be paid in designing them. The current *FireFly* UI is implemented using YUI from Yahoo!, the idea is to simplify the UI and keep it as similar is possible to widespread operating systems: the users list and every tool in *FireFly* is rendered within a sort of windows exposing traditional controls like drag and drop on title bar and maximize/minimize icons, in the upper part there is a status bar that mimics the feature of the status bar available under most common operating systems (see Figure 2).

3.4 System life cycle

Using AJAX for implementing an highly interactive system like *FireFly*, means to carefully design the policy of distribution of the updates among the users. The typical user activity is to append sentences in a chat session; because of the architecture of the browser, when a user clicks the submission button for a new contribution, the text is immediately sent to the server that has to bounce it to every other client.

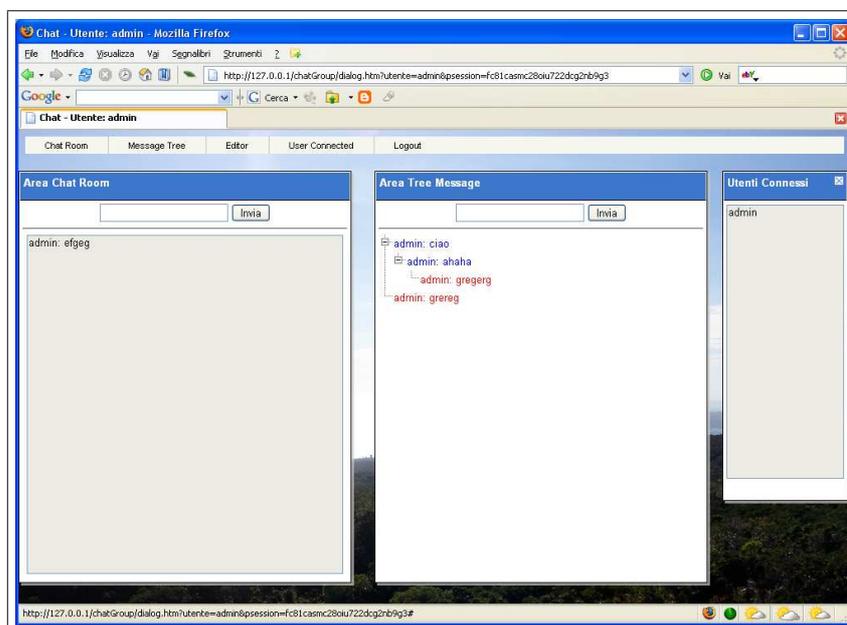


Fig. 2. A screenshot from the application.

Bouncing automatically newly available updates to the clients is not possible because the data exchange between the web server and the clients can happen just when the clients explicitly request for it. Using AJAX (and `XMLHttpRequest`) the client's browser is capable of *periodically request* updates and visualize them in the various tools. This is the key of the use of AJAX, these periodic updates cannot be avoided because of the HTTP protocol that is based on request/response mechanism [14], and the respond, that is the updates from other clients, can be sent just after a request generated by the browser. In figure 3 is shown the data exchange between the client and the server, continuous lines indicate the periodical updates, while dotted lines indicate the updates sent from client to server on every new contribution from user.

The frequency of such requests is a critical issue, too frequent requests create an heavy load on the server, while less frequent requests cause the slowdown of the interactions between users. In early testing we have used a 3 seconds interval, it is clear that this interval depends on various factors: the number of clients, the number, the size and the frequency of contributions from users. The server collects casual updates from clients

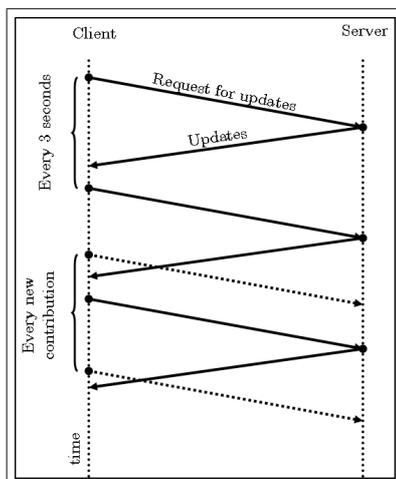


Fig. 3. Data exchange between client and server. lines between client and server indicate periodically updates, while dotted lines indicate casual updates. All the exchanged data are well-formed XML files.

and append them to the XML file that the periodical updates will request for.

4 Conclusion

The system we have designed is currently under development, our effort is in tidying up the code in order to make it modular. Our ambition is to design a modular system that would allow to developer to create tools on their own and just plug them into the system. One of the first step in this tidying up phase is to keep the XML exchange across the network the more efficient is possible, and this will request some ad-hoc measures of both generated traffic and parsing effort for both client and server.

Discussion As a summary we report here how we matched the prefixed goals reported in early sections:

- Easiness in the installation and management: *FireFly* needs just a one-step installation on the server. The installation of the web server is enough easy to be handled by common users and will not require complex settings.
- Low-end hardware: the system can be used exploiting pre-existent web server. In case no web server is available a small footprint solution can be employed.

- No Internet connection required: there is no-need for an internet connection. *FireFly* works on a LAN.
- Need for a cross platform solution: the capability of being cross-platform is achieved on both the sides, client and server, as a result of using AJAX technologies. Under various operating systems there exists plenty of web servers suitable to run *FireFly*. Whatever is the client operating system there will be a browser capable of executing the *FireFly* Javascript code.

Further undergoing developments are toward implementing new tools, like cooperative writing tool and graphical whiteboard.

References

1. Apache software foundation. <http://www.apache.org/>.
2. Dojo. <http://dojotoolkit.org/>.
3. Fastcgi. <http://www.fastcgi.com/>.
4. Gaztek ghttpd. <http://gaztek.sourceforge.net/ghttpd/>.
5. Gmail. <http://www.gmail.com>.
6. Html 4.01 specification: Forms. <http://www.w3.org/TR/REC-html40/interact/forms.html>.
7. lighttpd. <http://www.lighttpd.net>.
8. Sambar server. <http://www.sambar.com>.
9. script.aculo.us. <http://script.aculo.us/>.
10. Writely. <http://www.writely.com>.
11. Yahoo! user interface. <http://developer.yahoo.com/yui/>.
12. Youos. <http://www.youos.com>.
13. Jesse James Garrett. Ajax: A new approach to web applications. <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
14. H. Frystyk T. Berners-Lee, R. Fielding. Hypertext transfer protocol – http/1.0. 1996. <http://tools.ietf.org/html/rfc1945>.

A method for capitalizing upon and synthesizing analyses of human interactions

Annie Corbel¹, Jean-Jacques Girardot¹, Kristine Lund²

¹G2I / RIM, École Nationale Supérieure des Mines de St. Etienne, 42023 St. Etienne cedex 2

²I.C.A.R. (Interactions, Corpus, Apprentissages, Représentations), UMR 5191, CNRS, Université Lyon, École Normale Supérieure Lettres & Sciences Humaines, 15, parvis René Descartes, BP 7000 69342 Lyon Cedex, France

{Annie.Corbel, Jean-Jacques.Girardot}@emse.fr Kristine.Lund@univ-lyon2.fr

Abstract. It is often the case that analyses of human interactive activity are lost once an article is written about the results obtained. Although it is clear that corpora are gathered in order to answer particular research questions and that already collected corpora are often not adapted for answering new research questions, it is still interesting to reflect upon the capitalization and exploitation of analyses that have been carried out. For example, comparison of analyses, validation of analyses or alternatives modes of visualization could be possible. This article proposes a model of designation and extraction of parts of human interaction corpora using the *anchor* and *link* concepts that allow for experimenting on the reuse of analyses of human interactions.

Keywords: human interaction analysis, anchors, inter-coder reliability

1 Introduction

Many researchers are interested in the diverse forms of cognitive and social activities that take place when people interact together, for example, in teaching-learning situations or during cooperative problem-solving in the workplace. Computer Supported Collaborative Learning (CSCL) platforms, such as DREW¹ [1], [2] allow the researcher to collect and conserve computer-mediated interaction traces in the form of computer files. Researchers in the human sciences create other computer files when they transcribe (most often manually), the recordings of audio and video interactions. These two types of traces of human activity — issued from different sources — are the focus of analysis by researchers with particular objectives. Indeed a researcher will collect his or her data and thus define the type of trace, according to his or her research questions. As it stands today, these analyses, from which Ph.D. theses or articles are written, are not easily reusable and thus do not permit capitalizing upon analyses carried out for a given experiment or observed situation, or between different experiments or situations.

¹ Dialogical Reasoning Educational Website; see <http://scale.emse.fr/>

In this article, we address the possibilities of exploiting the analyses of traces of human interaction, for a single situation or across situations. The hypotheses we make and constraints that we recognize are the following:

1. The traces are available in the XML format, the semantics of which is known, at least informally. This is not a strong hypothesis: many CSCL tools directly produce such formats. In other cases, if the representation and the semantics of the traces are known, it is possible to convert them to XML format without loss of information.
2. The proposed approach does not prejudice the use of a specific tool or a prescribed format; it applies to the conjoint usage of different tools and methods of gathering traces, for example through one or more CSCL tools on the one hand, and by the manual transcription of audio or video, possibly with the help of an appropriate tool like [3] or [4], on the other.

One of the needs of the researcher in human and social sciences is to explore collected interaction traces in a pertinent and efficient manner, to annotate interesting phenomena and to obtain new documents that reflect the result of his or her activity. These new documents, represented in XML, will allow the comparison of these results within and across situations. Conversion into formats more appropriate for visualizing and disseminating results should also be made possible.

2 The form of human activity traces

In the context of previous projects (CESIFS², SCALE³, COSMOCE⁴), the authors carried out different experiments using the DREW platform [2]. DREW proposes different types of interaction (chat, whiteboard, argumentation grapher, text editor) and manages the creation of a trace (in XML) of the computer-mediated human activity that DREW makes possible. This trace is collected in the form of a sequence of events, each event corresponding to a single participant's intervention: a message sent in the chat, an element created in the whiteboard, an argument for or against a thesis put into the argumentation grapher, etc. In the document generated, these events are conserved in the order of their appearance, the DREW server arbitrating between events that are quasi-simultaneous.

In the context of the European project SCALE, a larger platform was developed called the Pedagogical Web Site (PWS [5] [6]). The PWS can replay in real-time a DREW session, carried out, for example, by two learners in a cooperative problem-

²The 'CESIFS' project (Conception et Etude de Sites Internet pour la Formation Scientifique) or Conception and Study of Internet Sites for Scientific Training), was supported by the French region Rhône-Alpes 1997-2000.

³The "SCALE" project (Internet-based intelligent tool to Support Collaborative Argumentation-based LEarning in secondary schools) was financed by the European Union "Information Societies' Technologies (IST) programme (IST-1999-10664) of the 5th framework between 2001 and 2004; <http://www.euroscale.net>, <http://drew.emse.fr>.

⁴The 'COSMOCE' project (Conception, Outils, Supports, Médias, Organisation pour la Collaboration des Entreprises) or Conception, Tools, Support, Media, Organization for the Collaboration of Companies, was supported by the French region Rhône-Alpes 2003-2006.

solving situation. It is possible to visualize the trace of their activity in html format and to perform analyses on the nature of their activity (cf. for example the Rainbow framework: [7]). Some of these experiments have also been the object of audio and video recordings, these recordings having been manually transcribed by researchers, in order to obtain documents that can be manipulated on a computer.

The traces that were gathered in the context of these projects were for the most part in XML format. However, if one takes into account the wide variety of CSCL tools and transcription conventions followed by researchers, it seems illusionary to attempt to propose a common transcription/trace format or even hope to define a kind of “pivot format” that can represent human activity, whether it is through an exceedingly complex format that expresses all the nuances and variations possible or whether it is through a simplified format that expresses a lowest common denominator. It is simpler and more reasonable to imagine that the XML trace documents are conserved, unchanged, in their original form, as the researcher chose to record them. Consequently, it becomes necessary to furnish the researcher with a tool that permits him or her to explore the collected corpus through a friendly interface. The minimal functionalities that should be supplied are:

- The visualization of corpus extracts;
- The possibility to annotate elements of the corpus;
- A search mechanism for the corpus.

Some of these functions can be provided with simple programming. Others necessitate the definition of a model of designation and extraction of parts of interaction corpora. It is this last point that we address in the method described in the following sections.

2.1 Analysis of computer-mediated human activity traces

Many researchers are interested in the *processes* that make up social and cognitive human activity in teaching-learning situations or during cooperative problem-solving in the workplace as opposed to being interested solely in a final common product that may be the goal of such situations. Thanks to the automatic chronological recording of human activity mediated by computer, researchers have the technological means since the 1990s [8] to respond to a variety of questions centered on *process*. For example:

- How do learners use the tools put at their disposal in relation to the activities they carry out? [9];
- What is the role of argumentation in the co-construction of knowledge? [10];
- How does structuring computer-mediated communication interfaces change the nature of interaction? [11];
- How do the internal factors of interaction (e.g. social talk) correlate with cooperative profiles (e.g. symmetry of roles) [12].

It is clear that each research question requires obtaining carefully chosen data that through specific analyses allow a response to be formulated. It follows that certain collected traces will not be adapted to addressing research questions for which the traces were not designed. For example, if a researcher is interested in how social talk relates to role symmetry, he or she would need to observe a task where roles can be

either symmetrical or asymmetrical. On the other hand, the gathering of this same data would not help him or her in answering a question pertaining to structuring communication, if in fact learners were given the same communicative interface or indeed if they were speaking unhindered, face to face. However, if the task generated argumentation and involved complex concepts, perhaps the trace would be interesting for studying the co-construction of knowledge, even though it was not originally designed for that purpose.

Despite the constraint of research questions guiding data gathering, and that as a consequence, already gathered data is not systematically adapted to new research questions, it is nevertheless interesting to stock analyses of corpora in a database in order to further exploit and capitalize upon them.

So, what then do we mean by exploiting and capitalizing upon analyses of interaction? Firstly, researchers from different disciplines or researchers using different methodologies have been known to work on the same corpus, see for example [13]. It is interesting to reflect on how one could facilitate the comparison of these different analyses, thus confirming comparable results obtained from different methodologies [14] or generating new research questions. Secondly, when the same analysis method is performed on many interactions by different coders, inter-coder reliability should be performed [15] in order to ensure that the coders agree on how to apply the coding scheme in question and thus guarantee the results and ultimately the coding scheme's replicability. Thirdly, it should be possible to automatically generate visualizations of specific analysis results by translating the corresponding XML documents into formats readable by other software applications.

In order to understand how such issues may be treated by the method proposed in this article, we illustrate an example analysis below, beginning with the Rainbow framework, used for analyzing computer-mediated pedagogical debates [7].

2.2. Taking the Rainbow framework further

The Rainbow framework was developed as part of the European SCALE project (see above) in order to analyze the restructuring of argumentative knowledge during computer-mediated debate [7]. In the context of the method proposed for this workshop, we illustrate how analysis of interaction corpora using Rainbow can be supported and how the analysis of argumentative interactions can be taken further.

There are seven categories within the Rainbow framework (hence the name): 1) outside activity not having to do with the task at hand, 2) social relation, 3) interaction management, 4) task management, 5) opinions, 6) argumentation and 7) explore and deepen arguments. We do not have the space here to further define these categories (but see [7] for a full description); rather we use Rainbow as an example of a coding scheme that can be applied to traces of computer-mediated human activity (cf. Fig. 1) and on which our proposed method of exploitation and capitalization can be applied.

46	10:26:47	Mark	ok let's argue	4. Task management
47	10:26:48	Mike	go ahead	4. Task management
48	10:26:49	Mark	ok	3. Interaction management
49	10:26:53	Nigel	i don't like solution C	5. Attitudes, opinions, agreement
50	10:27:16	Nigel	because we won't have a good driving force	6. Provide (counter-) arguments
51	10:27:22	Mark	ah really I like it	5. Attitudes, opinions, agreement

Fig. 1 An example of the Rainbow framework applied to an extract of computer-mediated human activity translated from data from the COSMOCE project.

Fig. 1 shows how each chat intervention may be categorized according to the Rainbow framework. It becomes clear that once different researchers have coded a number of different interactions making up a single corpus, it would be interesting to automate comparison of analyses in order to perform inter-coder reliability and obtain percentage of agreement on the whole corpus. In addition, other analysis methods can be applied to the same corpus. For example, in the COSMOCE project [16], after performing analysis with Rainbow, we further analyzed Rainbow categories 6 and 7 in order to ascertain the finer relations between arguing and how arguments are discussed within collaborative conception, precisely because Rainbow was not elaborated to analyze situations where design is the task (Fig. 2 illustrates the concept with a short extract).

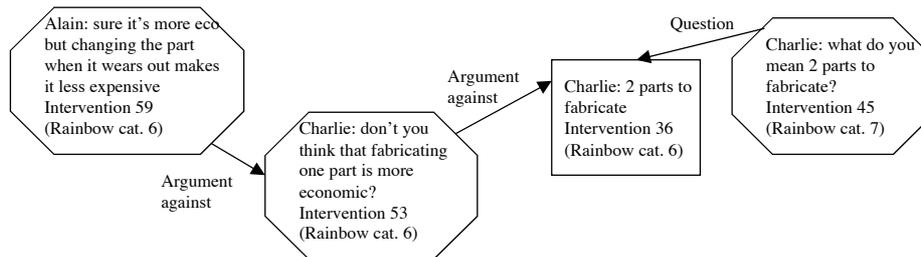


Fig. 2 illustrates a relational graph we produced that shows an example of the proposed relations between Rainbow categories 6 and 7.

In order to carry out this work, we needed to locate the chat interventions analyzed as Rainbow categories 6 and 7 in the original interaction and then propose semantic relations that existed between these interventions as a function of how we understood the designers to interpret their own discussion. We are currently developing a model of reasoning that describes argumentative activities of collaborating designers (cf. for example, [17] for a model of this type) for the situation we studied. We would like to perform these analyses on other interactions that have been analyzed by Rainbow in order to validate our model of reasoning.

The method we propose here (see the section below) is designed to support researchers in these kinds of undertakings: analysis according to a given coding schema, selection of analyses already done in order to perform further analyses, and finally comparison of analyses done by different coders or with different methodologies.

2.3 The proposed method

We begin by defining the term “primary corpus” (cf. [18] for an alternative definition) as the collection of all the documents gathered during the course of an experiment or observation. These typically consist of:

- Auditory or video documents that have been recorded during the experiment or during observation of the situation;
- Transcriptions of these recordings carried out by the researcher;
- Traces of computer-mediated interactions;
- Documents distributed to participants in the experiment/situation;
- Notes taken before, during and after the experiment/situation;
- All other documents judged to be pertinent by the researcher.

These documents are finite in number and will not evolve a posteriori, as they represent all the data gathered during and on the experiment/situation. In practice, we are interested in the documents that exist in computer format (having been originally generated in or translated into XML) for which an informal semantics can be defined.

We make the hypothesis that this primary corpus will be considered as fixed and unchangeable. All other documents created at a later date from this primary corpus will be an extract, a comment or an interpretation of the primary. Any annotations to the documents in this base will be expressed through an intermediary document (the “anchors document”) that will create references to the primary corpus. It could be the case that a study is performed on different primary corpora, these will be globally called an “observation base”.

The methodology described above allows us to constitute a corpus that contains all of the available data, without any information loss as no data is translated from one format to another. As mentioned previously, this corpus should be visualized and explored by the researcher. He or she should also be able to designate particular elements, annotate them and extract these elements or parts of them.

However, we cannot expect the human and social sciences researcher to master the different representations linked to specific software, even through the most friendly of XML editors available. We must therefore provide him or her with a tool that allows a visualization of the corpus he or she wishes to analyze.

Following an initial analysis of research practices, needs and existing tools, we propose the following tentative solution:

- The development of a generic browser, allowing for the visualization and the mark-up of the different documents that are part of the primary corpus.
- The development of an annotation tool, allowing for the linking of annotations to elements of the primary corpus.
- The development of an analysis tool allowing for the creation of links between elements of the corpus (a given chat intervention for example) and elements of the analysis method (for example, the task management category in the Rainbow method).

Documents pertinent to the analysis method (such as the enumeration of categories in Rainbow) constitute the *Analysis Base*.

Technological aspects

The use of XML [19] and the existence of related technologies allow us to list the specifications of these different tools.

Generic Browser

The use of formatting procedures for representing data contained in XML documents forms the basis of the Generic Browser. In our prototype, these procedures are written in Xquery [20], a language of interrogation and conversion, adapted to XML documents. Each particular type of XML document (for example a DREW activity trace) has an external file associated with it that describes which kinds of elements (in an XML sense) are considered as interesting by the researcher. A procedure for showing information (as defined by the researcher) is associated with such elements. It is the result of this procedure that is shown in the Generic Browser.

Mark-up Tool

The researcher in human and social sciences may at any time decide to mark up a specific element of the corpus. This mark-up process results in the creation of an *anchor*: a spatio-temporal designation of a corpus element. The anchor is an XML element that gathers diverse resources such as its type, a reference to a specific document in a primary corpus in the observation base, a geographical and/or temporal point in that document and complementary information (hour, date, author of anchor).

Each anchor is of a specific type, which describes how to interact with this anchor; this behavior is defined in `anchor-type` XML elements, where, for example, an XQuery expression describes how to display the anchor in the Generic Browser.

The collection of anchors is conserved in an independent document. This document can also be explored with the Generic Browser, thus allowing the researcher to immediately bring up the *anchored* elements.

Link Creation

A link is a simple XML structure, made up of a group of labeled anchors. Each anchor designates an element of the observation base or an element of a primary document. The label of an anchor is an identifier that indicates the role of the anchor within the link. Each type of link is described by a `link-type` XML element that indicates the set of anchors that can be put in the link and how these anchors can be validated, and describes how the link should be displayed in the Generic Browser. Here again, XQuery is used for validating and displaying information.

Annotation Tool

The annotation tool is a simple structured text editor that allows the researcher to create an annotation document in XML. Each annotation is represented by an XML element and is designated by an anchor. Annotating a corpus consists in to creating the desired textual information and building a link between this information and the part of the corpus that is annotated

In this way, an annotation can be represented by a link that contains:

- An anchor on the comment created by the researcher
 - An anchor (or more) on elements of the corpus
 - An anchor on the document describing the researcher him or herself and the general objective of his or her work
- (cf. Fig. 3 for an illustration of the relations between all the technological aspects described in this section).

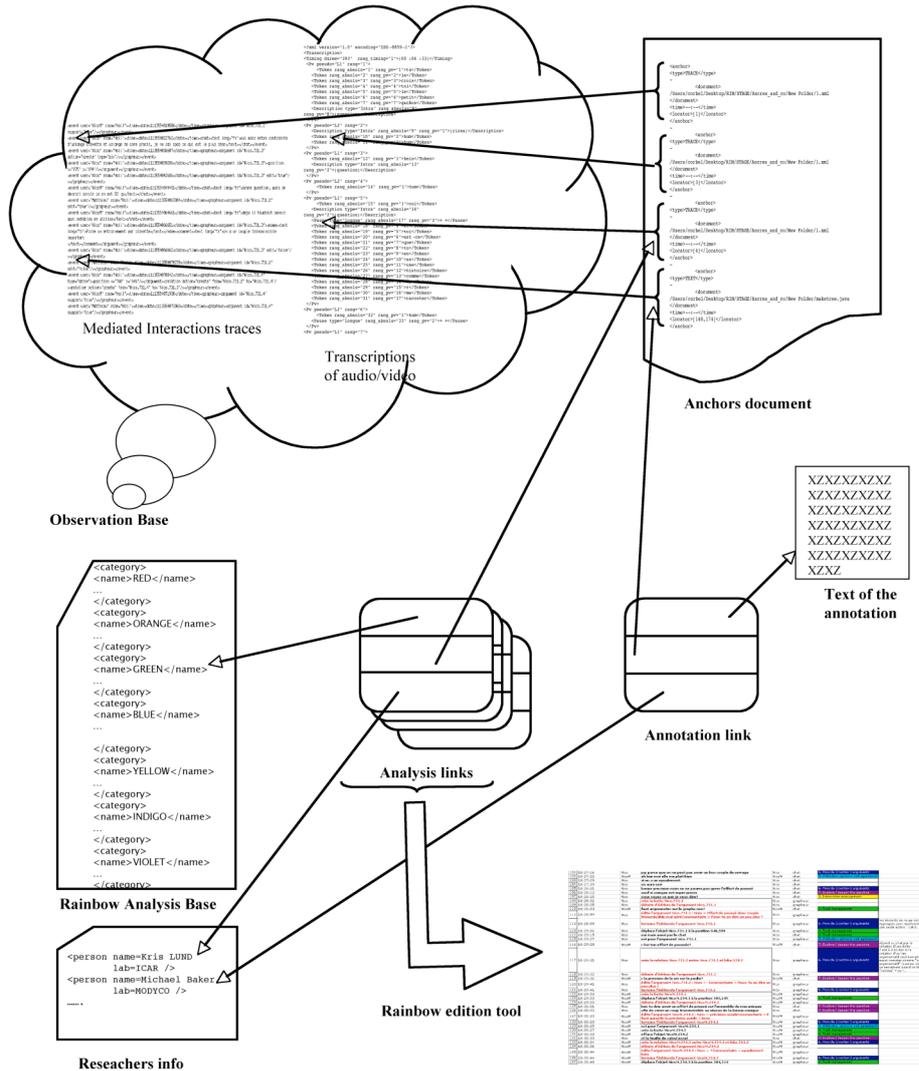


Fig. 3 illustrates the different components of the proposed method.

Analysis Tool

An analysis such as Rainbow (see the section *Analysis of computer-mediated human activity traces*) can thus be carried out with the help of the tools described above:

- The researcher can analyze the primary corpus by using the Generic Brower; he or she can create anchors on the elements deemed interesting;
- The researcher can also access an analysis base, a document in which the seven Rainbow categories are represented by anchors;
- It therefore becomes possible to place links between corpus elements and analysis categories.

The group of links thus created is in fact the analysis carried out by the researcher on the corpus. Once the analysis has been done, performing inter-coder reliability becomes straightforward. Analyses by different researchers on the same corpus can be compared and percentages of agreement calculated.

Since numerous kinds of computations and transformations can be performed on XML documents, the links resulting from an analysis can be used to provide useful representations of this analysis; XQuery procedures can be designed to generate a representation of the result of an analysis in Word or Excel format, or create inputs for a graph drawing software such as Graphviz [21] (used in fig 2).

Computations can also be performed to provide global perspectives, like the summary of activities of individual participants, time spent in specific tasks, etc.

3. Conclusions and perspectives

A model of designation and extraction of parts of human interaction corpora was proposed. An initial prototype has been built according to the proposed model and will firstly be tested on a selection of computer-mediated human interaction traces by researchers using the Rainbow framework. Next, we will develop a second analysis base, based on a different analysis method and test its use by researchers. Our ultimate goal is to provide an observation base of primary corpora that through the definition of anchors, allows researchers to annotate, analyze, validate analyses and visualize data using a single adaptive tool with provision for future reuse of the work done.

Acknowledgments. The authors would like to thank their colleagues from the COSMOCE project, the EAIH project and the European project LEAD for the inspiration for this work.

References

1. Quignard, M. (2000). Modélisation cognitive de l'argumentation dialoguée. Etudes de dialogues d'élèves en résolution de problème de sciences physiques. Thèse de doctorat de sciences cognitives. Grenoble : Université Joseph Fourier. [Cognitive modelling of argumentation dialogue. Studies of students in physics problem-solving].

2. A. Corbel, J.J. Girardot, P. Jaillon : DREW: “A Dialogical Reasoning Web Tool”, ICTE2002, Int. Conf. on ICT's in Education. Badajoz, Espagne, 13-16 Novembre 2002.
3. CLAN: <http://chilides.psy.cmu.edu/clan/>
4. TRANSANA: <http://www.transana.org/>
5. A. Corbel, P. Jaillon, H. Proton, X. Serpaggi : “A Guide for the PWS User, Scale Project”. Research Report G2I-EMSE 2004-400-009, December 2004, Ecole des Mines de Saint-Etienne, 18 pages. http://www.emse.fr/g2i/publications/rapports/RR_2004-400-007.pdf
6. A. Corbel, P. Jaillon, X. Serpaggi : “PWS : An open tool to construct, use and study pedagogical sequences in collaborative argumentation training” CSCL Conference and Workshop, Budapest - Számalk 20-21 February 2004.
7. Baker, M. J., Andriessen, J., Quignard, M., Amelvoort, M., Lund, K., Salminen, T., Litosseliti, L., Munneke, L. A framework for analyzing pedagogically oriented computer-mediated debates: Rainbow. Research Report IC-3. ICAR Laboratory, Lyon, France.
8. Jordan, B., & Henderson, A. (1995). Interaction analysis: Foundations and practice . The Journal of the Learning Sciences, 4, 39–103.
9. Bouvery, J. (2006). Activite et Usage Synchrones en Situation de Conception Collaborative à Distance d'un Texte Procedural [Activity and Synchronous Use in the Collaborative Design of a Procedural Text at a Distance]. Master's Thesis in Cognitive Science. University of Lyon, Lyon, France.
10. De Vries, E., Lund, K. & Baker, M.J. (2002). Computer-mediated epistemic dialogue: Explanation and argumentation as vehicles for understanding scientific notions. The Journal of the Learning Sciences, 11(1), 63–103.
11. Baker, M.J. & Lund, K. (1997). Promoting reflective interactions in a computer-supported collaborative learning environment. Journal of Computer Assisted Learning, 13, 175-193.
12. Lund, K., Rossetti, C., & Metz, Stéphanie (2006). Les facteurs internes à la coopération, influencent-ils l'activité médiatisée à distance? [Do internal factors of cooperation influence computer-mediated activity at a distance?] , M. Sidr, E. Bruillard & G.-L. Baron (Eds.) Actes des Premières Journées Communication et Apprentissage Instrumentés en Réseau JOCAIR '06, 6-7 Juillet, 2006, Université de Picardie Jules Verne : Amiens, pp. 310-329.
13. Traverso V., Détienne, F. (forthcoming), Méthodologies d'analyse de situations coopératives de conception, Nancy : PUN.
14. Salomon, G. (1991). Transcending the Qualitative-Quantitative Debate: The Analytic and Systemic Approaches to Educational Research. *Educational Researcher*, Vol. 20, No. 6 (Aug.-Sept., 1991), 10-18.
15. B. De Wever, T. Schellens, M. Valcke, H. Van Keer, (2005) Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education* 46(1), 6–28.
16. Cassier, J. L., Lund, K., Prudhomme, G., & Pourroy, F. (2006, 28-30 juin). L'argumentation comme support à l'analyse de la dynamique des connaissances dans une situation de co-conception synchrone à distance [Argumentation as support for the analysis of knowledge dynamics in a synchronous distant design]. Poster presented at the 17e journées francophones d'Ingénierie des connaissances, Nantes, France.
17. Conklin, J., & Begeman, M.L. (1989). Gibis: A tool for all reasons. *Journal of the American Society for Information Science*, 200-213.
18. Balthasar, L. Bert, M. (2005). La Plateforme “Corpus de Langues Parlée en Interaction” CLAPI, historique, état des lieux, perspectives [The Platform “Corpus of Spoken Language in Interaction”, history, state of the art and perspectives]. *Lidil : Corpus oraux et Diversité des Approches*, 31, pp. 13-33.
19. Extensible Markup Language (XML) : <http://www.w3.org/XML/>
20. XQuery 1.0: An XML Query Language: <http://www.w3.org/TR/xquery/> and <http://www.w3.org/XML/Query/>
21. Graphviz - Graph Visualization Software, <http://www.graphviz.org/>

Moderation of students' activities in co-located computer-supported collaborative learning using *Digalo*.

Einat Lotan-Kochan¹, Raul Drachman¹, Baruch Schwarz¹ & Amnon Glassner²

¹ School of Education, Hebrew University of Jerusalem, Israel

² School of Education, Ben Gurion University, Israel

Introduction

Digalo is a graphic-based software tool for supporting collaborative argumentation and structured discussion. It has been developed in the framework of the DUNES project (IST-2001-34153, <http://www.dunes.gr>). Using *Digalo* consists of synchronously co-creating maps built of written notes inside different cards (represented by diverse geometrical shapes), as well as using different arrows to represent various types of connections between the cards or contributions. These 'cards' and 'arrows' represent the ontology or the "grammar" of the discussion, which constrains but also facilitates and promotes the discourse [1,2] by guiding the learners to use specific speech acts and raising their awareness to their discursive roles, thus encouraging a certain type of discussion, such as critical dialogue.

Digalo in face-to-face settings

Although *Digalo* was designed to be integrated both in distance-learning settings and face-to-face settings, our experience in schools revealed that teachers and instructors prefer using it in face-to-face collaborative learning settings. We found that teachers use *Digalo* in three main ways:

1. for running an "opening discussion" as a first step of an inquiry process (e.g., brain storming, formulating and communicating opinions);
2. Co-constructing argumentative maps (in any stage of the learning process); and
3. for summarizing discussions (e.g. making group decisions, graphically presenting the structure of a problem/solution).

In other words, *Digalo* supports various types of face-to-face collaborative learning activities during the learning process. We also learned that it is of great importance to integrate *Digalo* activities in face-to-face collaborative learning settings, where the lesson design is as follows:

(A) face-to-face preparation activity (either teacher-led or small groups work);

- (B) *Digalo* activity;
 (C) face-to-face summarizing activity (teacher led and/or small groups work).

This design has proved to be most effective in terms of students' learning and structuring a whole inquiry process into one lesson unit. The oral face-to-face activities in A and C were found to contribute significantly to this learning.

During the last few years our research efforts focused mainly on students' learning and use of *Digalo*¹. However, we became more and more aware of the central role of the teacher or moderator in the implementation of the tool and its accompanying pedagogical method (argumentative-dialogue) on the one hand, and to the difficulties in facilitating students' group work in synchronous co-located *Digalo*-based discussions, on the other hand. These understandings led to a new research endeavor towards a better understanding of the teacher's role, in order to design a computational support for the moderation of *Digalo* activities. This is the focus of ARGUNAUT (IST-027728, <http://argunaut.org/>), a new R&D EU-funded project.

In this workshop we would like to reflect on the difficulties encountered by teachers in moderating *Digalo* activities, focusing on those related to it being a "networked-computing support for face to face [or co-located] collaborative learning situations". It is important to point out, again, that we see such difficulties as essential sources for further development and improvement of both the technological and the pedagogical aspects of *Digalo*'s implementation.

In a recent experiment, we found that teachers consider the following as key tasks (or roles) of the moderator of a *Digalo* activity:

1. Discipline and management of the activity,
2. Planning and organizing the lesson in which the *Digalo* activity is incorporated,
3. Encouraging participation,
4. Encouraging interaction and collaboration,
5. Presenting questions, asking for clarifications and explanations or playing the "Devil's Advocate" (in order to promote a dialectic argumentative dialogue),
6. Keeping the students focused on-task,
7. Emphasizing important contributions, aspects and ideas,
8. Making sure students use the ontology properly (i.e. encouraging an argumentative dialogue),
9. Providing technical support and making sure the application can be used easily in the classroom (steady infrastructure), and
10. Providing affective support and promoting students' motivation.

Obviously, these roles could refer to many face-to-face educational situations, and are not particularly unique to *Digalo*-based activities. What makes the difficulties encountered by these teachers in using *Digalo* in their instruction unique, then? We believe that the answer is in the integration of *Digalo* as a networked-computerized

¹ The results are reported in various publications of the DUNES project members (e.g. Johnson, Morgan & Simon [3]; van Diggelen, Overdijk & De Groot [1]; Glassner & Schwarz [2]).

tool for learning in a co-located environment. Such a setting means running a “double-mode” activity, where written-electronic interaction and spoken-oral interaction, take place at the same time. This means a heavier cognitive-load for the teacher or moderator of the activity – who has to follow both modes of interaction between students and within groups. Apparently, it is impossible to moderate students’ activity while monitoring both modes of interaction, and at the same time trying to fulfill any of the above mentioned roles.

Our group is currently handling this challenge on two planes: on the pedagogical plane, teachers developed two partial solutions:

1. Assigning a student the role of “head of the group”. These students received instructions as to how to handle their group’s work, but mainly – how to manage the group discussion using *Digalo* (roles #3 through #8, above). In some of the activities the group leader had to lead the group to make a final shared decision or reach agreement. The group leader was also responsible of presenting his/her group work to the whole class, in the summarizing activity.
2. Another partial solution can be found in the lesson design described earlier. Using this design the teacher can both guide the students (during the preparation activity) and get a limited picture of the groups work (in the summarizing activity).

On the technological plane, we are currently co-developing two possible solutions:

1. A moderator-assisting tool (ARGUNAUT) that will collect and process the data in real time and present them to the teacher in a way that decreases the cognitive load associated with the “double mode” interaction, by facilitating the monitoring of the written-electronic channel.
2. Another discussion-support tool – “Mapit” - is currently being developed within the KP-Lab project (IST-27490 (IP), <http://kp-lab.org>). This will enable both channels (written and spoken) to take place simultaneously through the “electronic mode”, hence reducing the need to split attention and other cognitive resources in monitoring the group work and interaction.

We expect this workshop will contribute further to these endeavors.

References

1. Diggelen, van, W., Overdijk, M., & De Groot, R. (2005). ‘Say it Out Loud in Writing’: A Dialectical Inquiry into the Potentials and Pitfalls of Computer Supported Argumentative Discussion, paper presented at CSCL2005, May 30 - June 4, 2005, Taipei, Taiwan.
2. Glassner, A., & Schwarz, B., B. (2005). The Role of Floor Control and of Ontology in Argumentative Activities with Discussion-Based Tools. Paper presented at CSCL2005, May 30 - June 4, 2005, Taipei, Taiwan.
3. Johnson, S., Morgan, A., & Simon, S. (2004). Report on DUNES Educational Value. Deliverable 6.1, restricted formal report.

Peer-to-peer Face-to-face collaboration

Delfina Malandrino and
Ilaria Manno

ISISLab
Dipartimento di Informatica ed Applicazioni “R.M. Capocelli”,
Università di Salerno, 84081 Baronissi (Salerno), Italy.
{delmal,manno}@dia.unisa.it

Abstract. In this paper, we present a *proof of concept* application of a technique that is designed explicitly for face to face collaboration software architectures. The objective is to minimize the impact on the installation and deployment of the application, that, while internally keeping a client-server architecture (in order to allow the centralize coordination and monitoring), presents to the user (both teacher and learners) as uniform work environment, integrating client and server components in one piece of software. In order to further limit the impact on the configuration, we define a *start and play* protocol, to start-up the application with no network configuration; the *start and play* protocol takes advantage from the particular conditions of the face to face context i.e. LAN setting. The application is built on the Eclipse core (Rich Client Platform), and inherits its plug-in based architecture and its advanced tailoring features.

1 Face2face collaboration systems

Current research in Computer Supported Collaborative Learning (CSCL) has produced many studies and several classifications of the situations where the collaboration takes place. The space-time matrix (see Fig.1) is a well-known classification [4] that defines the four basic space-time situations. A lot of works have studied the *different-time* AND/OR *different-place* situations to reduce distances (both in time and space), while there are fewer studies about the *same-time* AND *same-space* situations. Of course, existing synchronous systems for remote situations can also be used in the co-located situations, but the *same-time* AND *same-space* situations is substantially different from the remote ones and the technological support should take in account this difference. Indeed, the tools to support remote collaboration try to achieve a “virtual co-location”

	Same place	Different place
Same time	Co-located collaboration	Remote synchronous collaboration
Different time	Asynchronous collaboration	Remote asynchronous collaboration

Fig. 1. Space Time Matrix

enhancing remote communication by chat, e-mail, file sharing, audio and video conferencing, etc. In f2f situations, this kinds of communications channels are unnecessary because there is no distance to fill up. For these reasons, the systems to support co-located learning could and should focus on collaboration activities rather than on reducing distances, for example, they could provide reviewability and revisability [8], that are important characteristics in particular in the learning process [11].

Our team is involved as technical partner in the European project LEAD, in Sixth framework programme priority IST [5], whose goals are *to develop, implement and evaluate conceptual models, practical scenarios and associated networked-computing technologies for effective face-to-face problem-solving discussions*.

In this project we are focussing on design features and development solutions to produce a *face-to-face* (f2f) collaborative learning tool, in collaboration with others technical partner and according to the conceptual model outlined by the pedagogical partners.

In this paper, we approach the problem of designing an application for face-to-face collaboration that has minimum impact on the installation and management. We present an architecture and a small proof-of-concept prototype that was designed in order to test the effectiveness of our low-cost deployment strategy.

2 Software architectures

Most of the existing systems for CSCL have a client-server architecture. This model, in fact, simplifies data collection process and persistence management; furthermore, the client/server entities support the students/teacher roles, allowing to centralize in the server component the functionalities for the teacher, while the clients components offer the functionalities common to all the students.

The existing systems are Web-based, since the most are designed for remote situations (a survey is presented in [6]). These systems are not always suitable for a f2f didactic context, since they require to communicate with an external server (and therefore they require an Internet connection), and many schools employ restrictive firewalls and access policies. Furthermore, the teachers could not exercise fully control on the external server and is somewhat limited by its availability and configurability. On the other hand, some of the existing systems allow to install a local server, but the installation process is often too much complex for the end users, that may not have the experiences and capabilities to install and configure a Web server.

We aim to design a CSCL system explicitly applied to the f2f context, addressing the particular conditions of such context. In fact, in a co-located situation the system can use only the local area network, so it could and should do without external servers and Internet connection, in this way it can avoid many problems due to restrictive security policies, that are, often, commonplace in educational settings.

An existing cooperative system providing a LAN-based approach is MeetingWorks [7]: chauffeur and participant components are local applications and every participant links up with the chauffeur automatically, but the system¹ needs a shared directory to

¹ We have tested only the free version of the program, that has only LAN participants.

which every participant needs to gain access. Using a shared directory is a critical choice in the context of a classroom because the standard hardware and software equipment may not support many concurrent accesses (e.g. limitations on the number of simultaneous remote accesses in standard operating systems that are intended for desktop and not for servers). Therefore, the network use and configuration must be carefully designed not only to avoid problems due to security policies but to assure effectiveness and efficiency as well.

Beyond the network architecture, we are interested particularly in enhancing friendliness and deployment easiness: the system must be simple to configure, to start-up and to use, in order to encourage its usage and spreading.

In an overall view, we are designing a LAN-based system, providing a uniform work environment and a *start and play* protocol, to offer an application simple to install and to start up, in other terms, an application that exhibits a *low cost deployment*.

3 Our architecture

Several studies [9] suggest component-based architectures to address architectural requirements for collaboration systems. In particular, we are studying the Eclipse Platform [3] architecture. Eclipse is a component-based Integrated Development Environment that provides a framework (Rich Client Platform, RCP) to build general purpose applications using the Platform architecture. In the following we introduce briefly the Eclipse architecture (sec. 3.1), and then our approach to use RCP to build a face to face collaborative application (sec. 3.2).

Since the reasons (simplifying data management and matching teacher and students roles) to use the client-server model are well-grounded for the f2f system too, we do not set aside the client-server model, but we are studying how to use it in a *LAN-based* architecture, so that it can be independent of both Internet connection and external servers.

In order to simplify the system usage, we propose here an architecture with server and client components embedded in the same application, so that the system could provide a uniform work environment between teacher and students, and without requiring the management of a separate server. The idea is that the application looks *peer-to-peer* to the users even if their internal structure makes one of the peers (typically, the teacher's one) to be the server. Of course, this architecture leverage on the growing availability of CPU cycles on low-end desktops, and on the inherent limitation on the size of the classroom, which makes acceptable the workload on a server placed on a desktop machine.

Another aspect affecting user-friendliness is the start-up phase: to simplify the start-up phase we are defining a low cost deployment approach to allows the end users to *start and play* the collaborative application, with no network configuration. We describe these design features in the section 3.3.

3.1 Eclipse architecture

Eclipse is a component-based Integrated Development Environment grounded on three key concepts: plug-ins, extension-point and lazy activation.

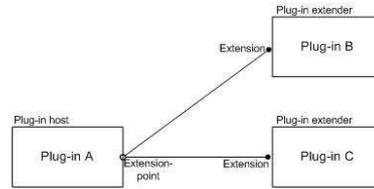


Fig. 2. The extender plug-ins B and C provide extensions to the host plug-in A.

A plug-in is the smallest independent software unit; even if a tool could be composed by more than one plug-in, the term plug-in is often used as “tool” or “component”. Every plug-in declares its identity and properties in a file *manifest*², so these information are available without activating the plug-in.

The extension-points define the rules of plug-ins composition: an extension-point is the point exposed by a plug-in to allow extensions from other plug-ins. The plug-in that exposes the extension-point is the plug-in host, while the plug-in that provides the extension is the plug-in extender (see fig. 2).

The plug-in host declares the extension-point in its file *manifest*, and the plug-in extender declares the extension in its file *manifest*, so that the information about extension relation between the two plug-ins are available without activating them.

The lazy activation is the property that allows to activate a plug-in on demand, so that there can be a lot of plug-ins installed but only few active.

Beyond the flexibility and scalability, the Eclipse architecture assures the *extreme tailorability* [2, 9, 10], allowing customization, integration and extension.

3.2 Building on Rich Client Platform.

Rich Client Platform (RCP) is the “core” of Eclipse: it is composed by the fundamental plug-ins, mainly to manage graphic interface and plug-ins life cycle, without any specific feature of the development environment. The RCP is a framework to build general purpose applications based on the Eclipse architecture (see fig. 3). The applications built on RCP inherit the tailorability provided by the Eclipse architecture.

To build the system on the RCP framework, we have to define the components of the application. We can distinguish two types of building blocks: the Core and the collaboration tools. Each component, the Core and the tools, is a plug-in. The Core provides fundamental functions, that are, at least, user awareness (presence and activity), installed collaboration tools discovery, start-up of tools (on demand, if possible), definition of the rules for composing the building blocks in the system.

The collaboration tools can provide any kind of functionality (free chat, structured chat, graphic shared editor, mix of previous, games, etc.); they must only observe the

² As a matter of fact, the *manifest* is a couple of files: *plugin.xml* and *manifest.mf*, that contain respectively information about relations with other plug-ins and about the runtime. They are often referenced as a single file, first for historical reasons and then because they can be edited with a single advanced editor.

composition rules fixed by the Core. The Core depends on RCP (see fig. 4) and is the main plug-in, that is, the plug-in defining the application. The Core provides an extension-point named `tools` defining the API that any collaborative tool must implement to be integrated in the system. This extension-point (like all extension-points) may have zero or more extensions. A plug-in extender has to declare in the file `manifest` an extension to the extension-point `tools` and has to implement the API specified by the extension point. The Core analyzes the extensions to the extension-point at runtime, so it is possible to add a tool to the system without changing the Core.

The plug-in based architecture allows to build each tool component with its own server embedded. The idea of a server for each tool has two reasons; first, in this way the Core ignores completely the tools details (and the tools servers details), so that whatever tool will be needed, it could be added without modifying the Core, since the tool embeds its own specific server functionalities; second, having a server for each tool and thanks to the lazy activation property, in each moment only the required tool servers are running. So, the strongly component oriented architecture of Eclipse assures fully tailorability, thanks to plug-ins and extension-point concepts. Furthermore, the *lazy activation* assure scalability: each collaborative tool will be activated only when required.

The flexibility and the extendibility of RCP would allows to extend the system as the collaboration needs arise, achieving a richer system, placed at the top of the classification framework presented in [10], where at the bottom there are basic collaboration functions, while at the top there are “comfortable” collaboration functions.

3.3 Low cost deployment: uniform work environment and start and play protocol

Part of our studies concerns the problem of the start-up: we would enhance start-up transparency so that the users could start the application and could use it with no configuration (i.e. *start and play*). Furthermore, we aim to provide a uniform work environment to make semi-transparent the difference between server and clients: they are integrated in the same application so that the application server instance is not perceived

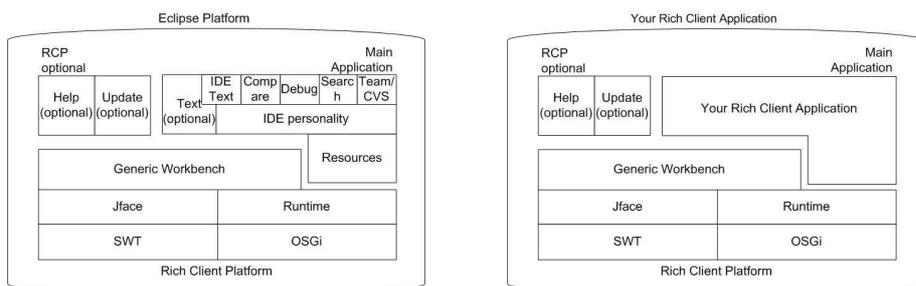


Fig. 3. (a) The Eclipse architecture and (b) a general purpose application on RCP

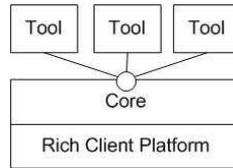


Fig. 4. The Core based on RCP can be extended by many tools.

by users as “the server” but as “a powered peer” (“powered” because it has more functionalities than standard clients). From technical point of view, this is merely a seeming difference, but from user’s point of view there is no external application to install, to configure, to start up and to manage. For these reasons, each component contains both the server side and the client side, even if, in each moment, only one instance of the application over the net runs in the server mode.

To achieve a *start and play* application, we developed a UDP-based server localization protocol, using only the local network. When the application starts, it is in client mode, and the Core client sends a “server lookup” message in broadcast; if in a timeout it does not receive the server reply, it instances and runs the Core server. Every subsequent application sending the “server lookup” message will find the server (see fig.5). Furthermore, the Core manages the start-up of the tools, so when a user (see fig. 6) requires to start a tool, the core client of user 2 sends a “start tool” message (specifying the id of the tool) to the Core server. When the Core server receives the “start tool” message, it instances and runs the tool server and forwards to all users the “start tool” message; each users that receives the start message runs the tool client. Each tool client sends a “tool server lookup” in broadcast, and will receives the reply of the tool server.

Since in CSCL f2f systems it is desirable that some operations are reserved to the teacher, the servers should be hosted by the teacher. To match this requirement, we have defined a *client running mode* and a *server running mode*. The running mode can be explicitly enabled by specifying a command line parameter. The teacher application

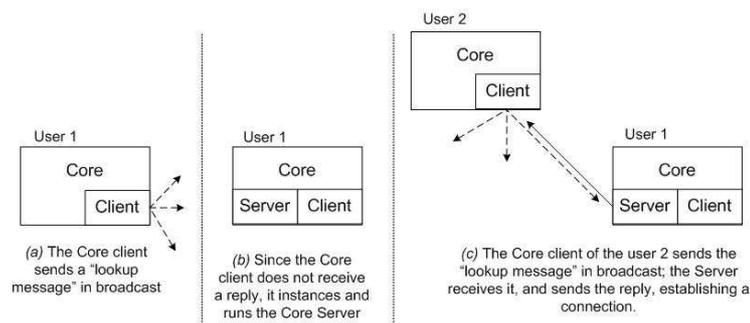


Fig. 5. The Core activation sequence.

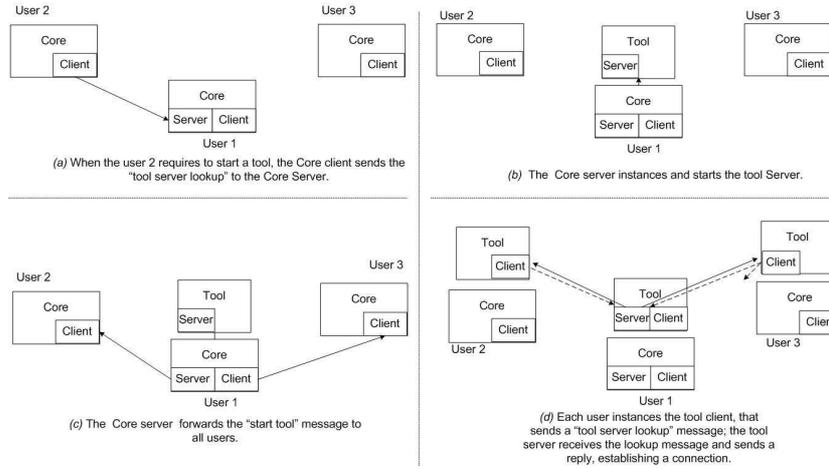


Fig. 6. The tool activation sequence.

instance runs in the server mode and directly creates and starts the Core server, skipping the lookup message broadcast; the students application instances run in the client mode and look for the server, but if do not receive reply, do not instance nor start the server. Obviously, if there is an application instance in server mode, all the other must be in client mode (or, however, they must fail if try to instance a server). This solution keeps the uniform work environment and the start and play phase, but reserves the access to the servers functionalities for the teacher. As matter of fact, we have forced (by the running modes) the protocol to achieve a powered application instance for the teacher, because the original protocol does not impose conditions about the user that hosts the server. Indeed, with the original protocol, the first user starting a component, instances and runs the server of the component (*p2p running mode*), so that servers of different components could be hosted by different users, moving the system toward dynamic architectures [1]. This solution did not seem suitable for educational settings, where it is preferable to instance and run all the servers at the teacher application, to provide servers functionalities access only to the teacher. The extreme tailorable architecture of RCP enhances the start-up transparency allowing to design the system with a set of servers: a server for the core and a server for each provided tool³. With the plug-in based architecture and the lazy activation it is possible to design the system so that in each moment a chosen tool can be activated and then, silently, the server of the tool is started and then the clients of the tool find it.

These features together (transparent start-up and uniform work environment) provide the end user with the perception of a peer-to-peer system, although the system is instead a client server one.

³ In the educational setting all the server are hosted on the teacher application instance, but even so, it is preferable having a server for each component because this layout enhances extensibility (see sec. 3.2)

4 Conclusions and future work

Here we have presented our studies about the architecture of a system designed and developed explicitly for face to face collaborative learning. Our system provides a uniform work environment and allows the users to *start and play* the application. Compared with existing systems⁴, our system is simpler to install, to start-up and to use, because it has neither separate server to manage (uniform work environment) nor network configuration to execute (*start and play*). Furthermore, it inherits from Eclipse advanced tailoring properties.

Since our system has to address specifically face to face collaboration, we can utilize the particular conditions of such context to achieve a more friendly application. The *start and play* protocol takes advantage of the LAN-based context, and really, it is workable only on wired-LAN, because the UDP broadcast is often disabled out of the LAN. Furthermore, the local network often offers low variance delay and this helps to prevent (but it is not the best solution, of course) race conditions in the server start-up phase. Vice versa, the high delay variance of wireless LAN may cause anomalous behaviors of the protocol, due to, for example, expiring timeouts. To use a similar protocol on wireless LAN, it must be specifically designed to address WiFi peculiarity.

In the context of face to face collaborative learning, the server functionalities should be managed by the teacher, so that all the servers (Core server and tools servers) are hosted on the same application instance (the teacher's one). Actually, the described architecture forces this behavior, but interesting studies concern the p2p running mode, that provides the opportunity of hosting the servers in a distributed way over different application instances (for example, the first user starting a tool can host the tool server), migrating the system toward dynamic architectures [1]. Even if the p2p running mode may be unsuitable for educational settings, we wish briefly describe some interesting features and problems related to the p2p running mode. The opportunity to have a distributed servers set allows to share the workload between all the users; furthermore, this allows to relax the roles strictness, matching situations more dynamic and flexible (such as a work group where different members have different roles and competences) than the educational one, where there are the well defined student and teacher roles. A problem related to the distributed servers set concerns the shutdown of a single application instance hosting one server: it should be a *controlled* shutdown, to allow the server migration toward another application instance (i.e. another application instance creates and runs the server). An even more complex problem concerns the *crash* of an application instance hosting one server, and this requires further studies, as well as the data management protocol. Obviously, the p2p running mode and the distributed servers set are based on the idea that each component embeds its own server. Maybe it is too early to make statement about the user capabilities required by a system with a distributed server set, although we expect that the distributed architecture has no consequences on the user level.

⁴ i.e. existing remote systems used in a face to face context

References

1. G. Chung and P. Dewan. Towards Dynamic Collaboration Architectures. In *Proc. of CSCW'04*, pages 1–10.
2. A. Dimitracopoulou. Designing Collaborative Learning Systems: Current Trends and Future Research Agenda. In *Proc. of Computer Supported Collaborative Learning 2005: The next Ten years!*, 2005.
3. Eclipse. <http://www.eclipse.org>.
4. Robert Johansen. *GroupWare: Computer Support for Business Teams*. The Free Press, New York, NY, USA, 1988.
5. Lead - technology-enhanced learning and problem-solving discussions: Networked learning environments in the classroom, 6th Framework Programme Priority IST. <http://lead2learning.org/>.
6. J. Lonchamp. Supporting synchronous collaborative learning: a generic, multidimensional model. *International Journal of CSCL*, 1(2), to appear in June 2006.
7. MeetingWorks Software. http://www.entsol.com/html/meetingworks_software.html.
8. G.M. Olson and J.S. Olson. Distance Matters. *Human Computer Interaction*, 15(2-3):139–178, 1994.
9. H. ter Hofte R.Slagter, M.Biemans. Evolution in Use of Groupware: Facilitating Tailoring to the Extreme. In *Proc. of Seventh International Workshop on Groupware*, 2001.
10. A. Sarma, A. van der Hoek, and L.T. Cheng. A Need-Based Collaboration Classification Framework. In *Proc. on Eclipse as Vehicle for CSCW Research, Workshop at CSCW 2004*.
11. R.J Simons. Three ways to get content based and in-depth conversation in on-line learning: revisability, focussing and peer feedback, 2006. Seminario Tecnologia Cultura e Formazione.

Technology Appropriation in Face-to-Face Collaborative Learning

Maarten Overdijk and Wouter van Diggelen

Utrecht University, Research Centre Learning in Interaction,
Heidelberglaan1, 3584 CS Utrecht, The Netherlands
{m.overdijk, w.vandiggelen}@fss.uu.nl

Abstract. Studies in computer supported collaborative learning frequently under-expose the interaction between students and technology. To gain more insight in the way student groups interact with educational technology, we examine how students ‘appropriate’ this technology. The notion of technology appropriation implies a process of social construction in which the actions and thoughts of the user are shaped by the technology, while the meaning and effects of the technology are shaped through the users’ actions. In this paper, we develop a problem analysis from theoretical findings, and work towards an initial, tentative concept of technology appropriation.

Key words: Collaborative learning, Technology appropriation, Discussion support, Face-to-face discussions.

1. Introduction

The field of computer-supported collaborative learning (CSCL) advocates the deployment of technology to promote specific interactions between learners that lead to collaboration and learning. Software tools that support discussion within student groups, for example, may facilitate a free exchange of ideas, argumentation or critical thinking within the group. These tools influence group behavior by triggering certain actions, and by shaping interactions between the students, in a way that has a positive effect on the discussion.

In addressing how collaborative learning within student groups is influenced through the use of technology, CSCL research frequently under-exposes the interaction between the students and the technology. The technology is predominantly treated as a variable with a stable influence on the thoughts and actions of the students. This influence is assumed – often implicitly – to be independent of the students’ actions. There is evidence to doubt the ground of this assumption. Essentially, because the assumption reflects a deterministic view towards technology use, and conflicts with the premises of socio-constructivist theory. Questioning of this deterministic view has led to several ‘emergent perspectives’, which propose that the use and effects of a technology emerge on the basis of complex social interactions among users [4].

To gain more insight in the way student groups interact with technology, we examine the students' 'appropriation' of the technology. We claim that the use and effects of a technological tool emerge from the interaction between user and tool, based on a mutual influence between them. The notion of technology appropriation implies a process of social construction in which the actions and thoughts of the user are shaped by the technology, while the meaning and effects of the technology are shaped through the users' actions.

If we aim to understand the potential of educational technology to enhance certain processes in the collaboration between students, then we need to be informed about the dynamics of student groups interacting with technology. The notion of technology appropriation, as we will point out, has important implications within the LEAD¹ project, and also within the broader context of CSCL research.

2. The LEAD project

Problem solving discussions are common in classrooms where learners work collaboratively on a task. Groupwork, for example, requires that learners discuss in order to make progress and to succeed as a group. Learners have to share their understanding of a problem and explore different directions to solve it. They have to explicate their line of reasoning, provide arguments, and reach conclusions that are sufficiently shared to proceed with the task. While doing all this, they may have to deal with conflicts and differences in opinion. Due to the challenging character of a problem solving discussion, learners do not always succeed in making the most of it.

The objectives of the LEAD project are to develop and evaluate conceptual models, a didactical method, and accompanying network-computing support to enhance problem-solving discussions in face-to-face classroom settings. The project team posits the claim that the quality of group processes and outcomes can be enhanced through the combination of face-to-face and computer-mediated communication.

The pedagogical research of the various project partners can be positioned within one of three interrelated themes: interaction between students, interaction between students and an external artifact, and the evolving situation. The research proposed in this paper can be placed in the second theme, interaction between students and an external artifact. From the perspective of this theme, the goals of the research are: (1) to formulate a conceptual model of technology appropriation, (2) to study empirically the way in which a network-computing technology is appropriated by face-to-face groups during problem-solving discussion, and (3) to inform the technological and didactical design process in the LEAD project.

¹ The LEAD project is funded within the Sixth Framework Programme of the EC (2nd call on Technology Enhanced Learning).

2.1 Problem statement

The method and technology that are being developed in LEAD help groups to move forward in their discussion by triggering certain actions, and by shaping interactions among the group members.

The method comprises a designed sequence of activities that fit with certain pedagogical objectives. It specifies certain elements in the educational environment, like the task, instructions, and certain rules and techniques for collaboration. The method furthermore specifies the deployment of the Discussion Support System (DSS), that is, when it is deployed, why, and under which configuration. Part of the method will be reflected in the structures of the technology.

The DSS will present students with two types of tools: a text-based conferencing tool and a graphical, shared workspace tool. The tools trigger certain actions from the students, for example by making a notation system available. The notation system promotes certain communicative acts, like providing arguments or asking questions. Shaping of interactions is achieved through the use of certain techniques or rules that are reflected in the structures of the tool. For example, the tool enables students to participate simultaneously in the discussion. The process structure of the tools guide the content, pattern or timing of the communication [14].

The design of collaborative technology is based on hypothesis about how artifacts shape cognition and collaboration [19]. Similarly, the DSS is designed on basis of theory and hypothesis about how small-group problem-solving discussions between students proceed, and how they could be enhanced through technology. As we point out below, it is a goal of this research to develop and test such hypothesis.

To some extent, the tools reflect a certain intention to the user regarding their capabilities and how they should be used. Intentions about their use and effects are also explicated through the instructional design or ‘script’ that the tools are a part of when they’re presented to the student group. However, we argue that the students not necessarily use the technology in accordance with these intentions.

The effect of any didactical intervention that incorporates technology depends in part on a process of technology appropriation. Having said this, we ask the following research questions: how should we conceptualize technology appropriation; and how does technology appropriation affect the students’ discourse?

2.2 Research and development strategy

The research will follow LEADs’ research and development strategy [15]. In short, we formulate a descriptive conceptual model based on preliminary theoretical and empirical findings that were obtained through a problem analysis. We then proceed with an empirical evaluation of this model, which we subsequently adapt according to our findings. This results in a prescriptive model that may inform both the didactical and technological design process, that is, the development of the Discussion Support System and the didactical method.

In a general sense, the research may contribute to our understanding of the role of computer technology in face-to-face problem-solving discussions.

Throughout the remainder of this paper we develop a problem analysis from theoretical findings, and work towards an initial, tentative concept of technology appropriation.

3. Problem analysis

The conceptual part of our research question deals with defining what technology appropriation is, and how we should model it. Literature review brings different aspects of ‘appropriation’ to the fore that may be of relevance to our study. We focus on the notion as it can be found in socio-cultural theory, and with socio-constructivist perspectives on the use of technology. In our view, these two theoretical strands develop different aspects of the notion, which make up a comprehensive picture when combined. Of central importance herein is our conception of the relation between user and technology.

3.1 Appropriation of cultural tools

The notion of appropriation has been frequently used in socio-cultural learning theory. Rogoff [13] uses the term ‘appropriation’ to refer to “the process by which individuals transform their understanding of and responsibility for activities through their own participation”. What is appropriated in this case are cultural tools, like language, procedures or ‘technical tools’ (e.g. a technology) that are attached to a particular practice. For Wertsch[18], ‘appropriation’ of a cultural tool can be distinguished from ‘mastery’ as a form of internalization. Whereas ‘mastery’ refers to “knowing how to use a mediational means with facility”, ‘appropriation’ refers “to taking something that belongs to others and making it one’s own” [18]. This should not be read as taking ownership of something, but rather as adapting it to one’s own use. Both Wertsch and Rogoff refer to Bakhtin in their use of the term. According to Bakhtin, a speaker appropriates a word when he adapts it to his own semantic and expressive intention [1].

Human thoughts and activities undergo a transformation when they become mediated by cultural tools. The notion of mediation in the socio-cultural approach is to a large extent inspired by cultural-historical psychology [16]. Basically, tools are created and transformed during the development of an activity and carry with them a particular culture - the historical remnants from this development. The use of tools is a means for the accumulation and transmission of cultural knowledge. This developmental process influences the nature of external behavior and also of the mental functioning of individuals [2]. This emphasis urges to go beyond the here-and-now interactions of tool-users; the interaction process should be incorporated in a broader cultural and historical frame of reference. However, a ‘technical tool’ like a collaborative technology may also evolve ‘here and now’ over a short period of time.

3.2 Social shaping of technology

In a different strand of theory, the notion of appropriation has been used to explicate a mechanism through which technology is socially shaped [5, 9, 3].

Carroll et al. [3] define appropriation as a process in which a technology is explored, evaluated and adopted or rejected by users. According to their view, users make use of certain capabilities of a technology, and reject others, in order to satisfy their needs. They see appropriation of mobile technologies by young people as a result of the interplay between what people desire, the capabilities and implications of the technology, and the situation of use [3].

DeSanctis and Poole [5] use a different concept of appropriation. Their use of the term can be traced back to Ollman, who defines appropriation as constructive utilization [10]. According to them, ‘appropriations’ of a technology are immediate, visible actions that evidence deeper structuration processes. Agents appropriate rules and resources that become available as groups interact while using advanced information technology [5].

Technology appropriation can be described as a process that takes place on different levels of social organisation, that is, on the level of the individual user, a group of users, or on the level of the larger sociocultural environment. Carroll et al. [3] place appropriation on the level of the individual user. DeSanctis and Poole [5] conduct an institutional analysis, and define appropriation on the level of the organisation.

The socio-constructivist approaches to technology focus on the fact that technologies are socially shaped, and that their use and effects depend on human contingencies. This perspective suggests that a technology gets its form and meaning in-interaction. The technology-in-use is not a stable artifact with fixed characteristics that are independent from practice. In stead, students construct essential characteristics of the tool when they work with it. It follows that technology is not necessarily used in accordance with the designers’ intentions. “Technological artifacts, in both their form and their meaning, are socially shaped, as opposed to being the clearly defined products of particular inventors or innovators” [7].

3.3 The relation between user and technology

In previous research we described the mutual influence and dependency between students and technology [11]. Several aspects about their relation remain unresolved.

We can make different assumptions about the relation between the students and the technology. One assumption would be to state that the behavior of the students is directed through features of the technology. Another assumption would be to state that the students need to actively explore the technology and make conscious choices in order to achieve a desired outcome.

The theory of affordances [6] fits with the first assumption. Scholars have argued in favour of the notion of ‘affordance’ as an analytic tool to analyse the ‘effects’ and ‘constraints’ of a technology [7, 8]. The affordance of an object refers to the possibilities for action carried by this object. The concept originates from environmental psychology, and is closely related to theories of perception. Main shortcomings of the theory are that (1) it describes explanations on the level of the individual, and says little about the group level, and (2) leaves little room for a process of mutual shaping, since the affordance is considered invariant. As Gibson states “The affordance of something does *not change* as the need of the observer changes. The observer may or may not perceive or attend to the affordance, according to his needs, but the affordance, being invariant, is always there to be perceived” [6].

The theory of ‘adaptive structuration’ (AST) fits with the second assumption. DeSanctis and Poole [5] describe how people utilize technology in organizations, and how their interaction with the technology influences their social practice. In AST, agents are knowledgeable, and have a conscious influence on the course of events. “New social structures emerge in group interaction as the rules and resources of the technology are appropriated in a given context and reproduced in group interaction over time” [5]. The concepts they use illustrate a construction process that transcends notions of ‘internal’ and ‘external’. Consequently, any change or transformation that takes place cannot be solely located within a subject, the object – in this case the technology – undergoes changes also.

4 Tentative concept of technology appropriation

Technology appropriation occurs when someone puts into use a technology in a goal-directed activity while the properties of the technology, and the acts required to accomplish the goal by means of the technology, are unacquainted. When someone is presented with a new technology, he or she appropriates this tool by ‘adapting’ it in a goal-directed activity. That someone has to make sense of the properties of the tool, and find ‘a way of doing’ to perform the activity. When a group of people is presented with a tool, technology appropriation occurs on the level of the group. In this case, the group has to make sense of the technology, and adapt it in a joint activity.

Technology makes certain rules and resources available, and it provides opportunities for interaction that would be hard to achieve without the technology. However, technology appropriation does not simply refer to acquisition of knowledge about an object, or to ‘learning how to’ do or apply something with the technology. Appropriation of a technology simultaneously transforms user and technology. It does not only cause change in the knowledge and skill of the user, but it also causes change in the properties of the technology. Central to the concept of appropriation is a mutual shaping. The concept implies a process of social construction in which the actions and thoughts of the technology user are shaped through the use of the technology, while at the same time the meaning and effects of the technology are shaped through the users’ actions.

The educational environment in which the students work encompasses more than the technology, the didactical method and the other members of the group. Elements on different levels of social organisation are of influence on the classroom practice. One could think of, for example, the routines of the practice, or norms that prevail on the level of the educational institution, which can be of influence. One could also think of the history of a student group, or the experience of the individual student. Processes that take place on each of these levels of social organisation may influence the way the technology is brought into action, and affects the students' discourse.

5 Technology appropriation and the students' discourse

The empirical part of our research question deals with how technology appropriation affects the students' discourse.

When the students lack an understanding or are confronted with a technology that seems confusing, they may engage in a process of sensemaking [17]. The use and effects of the technology emerge from the interaction between the students and the tool, as a result of interrelated individual and group processes. We make an analytical distinction between the user and the external artifact, and separate between individual and group processes. We focus on the 'here and now' of the interaction process, which may evolve over short time periods.

In previous research [11] we studied technology appropriation by examining basic actions that students performed in a graphical, shared workspace tool. This research indicates that individual students make certain choices during the process of appropriation. The study revealed that students make different choices both within and between groups. The choices they made influenced the effect of the tool, and led to differences in their discourse. In the planned continuation of this research we want to examine how the students make sense of the features of the tool, and find a way of working, as a group.

The specific focus of LEAD, the combination of face-to-face and computer mediated communication, provides some opportunities to learn more about technology appropriation. This situation enables us to investigate to what extent the students make use of (a) observations; (b) talk; (c) gestures, or (d) actions in the tool during their appropriation of the technology.

References

1. Bakhtin, M. (1981). *The Dialogic Imagination: Four Essays*. Austin: University of Texas Press.
2. Bannon, (1997). Activity theory. Retrieved from: <http://www.sv.cict.fr/cotcos/pjs/TheoreticalApproaches/Activity/ActivitypaperBannon.htm>

3. Carroll, J., Howard, S., Vetere, F., Peck, J. and Murphy, J. (2002). Just what do the youth of today want? Technology appropriation by young people. Proceedings of the 35th Hawaii International Conference on System Sciences.
4. Contractor and Seibold (1993). Contractor, N.S. and Seibold, D.R. (1993). Theoretical framework for the study of structuring processes in Group Decision Support Systems: Adaptive Structuration Theory and Self-Organizing Theory. *Human Communication Research*, 19(4), 528-563.
5. DeSanctis, G. and Poole, M.S. (1994). Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory. *Organizational Science*, 5(2), 121-147.
6. Gibson, J.J. (1979). *The ecological approach to visual perception*. New Jersey: Lawrence Erlbaum.
7. Hutchby, I. (2001). Technologies, Texts and Affordances. *Sociology*, 35 (2).
8. Kirschner, P.A. (2002). Three Worlds of CSCL: Can we Support CSCL? Heerlen: Open University of the Netherlands.
9. Mackay, H. & Gillespie, G (1992) Extending the Social Shaping of Technology Approach: Ideology and Appropriation. *Social Studies of Science*, 22(4).
10. Ollman, (1971). *Alienation: Marx's conception of man in capitalist society*. Cambridge University Press.
11. Overdijk, M. & Van Diggelen, W. (submitted 2006). Making sense of tools and artifacts: the student-tool connection. Submitted to: *International Journal of Computer-Supported Collaborative Learning*.
12. Palincsar, A.S. (1998). Social Constructivist Perspectives on Teaching and Learning. *Annual Review of Psychology*, 49, 345-375.
13. Rogoff, B. (1995). In J. V. Wertsch, P. d. Rio, & A. Alvarez (Eds.), *Sociocultural studies of mind* (pp. 56-74). New York: Cambridge.
14. Tyran, C.K. & Shepherd, M. (2001). Collaborative Technology in the Classroom: A Review of the GSS Research and a Research Framework. *Information Technology and Management*, 2, 395-418.
15. Van Diggelen, W. & Overdijk, M. (2006). Network computing support for face-to-face collaborative learning. Proceedings of the EC-tel conference, Crete 2006.
16. Vygotsky, L. S. (1978). *Mind and Society*. Cambridge, MA: Harvard University Press.
17. Weick, K.E. (1995). *Sensemaking in Organizations*. Sage: Thousand Oaks, California.
18. Wertsch, J.V. (1998). *Mind as action*. Oxford University Press.
19. Woods, D.D. (1998). Designs are hypothesis about how artefacts shape cognition and collaboration. *Ergonomics*. 41:2, 168-173.

TEACHERS PERCEPTION OF COMPUTER SUPPORTED PROBLEM SOLVING: AN ITALIAN RESEARCH

Luca Tateo¹, Giuseppina Marsico¹, Maria Annarumma¹, Giovanna Ammaturo¹,
and Antonio Iannaccone¹

Department of Education Sciences, University of Salerno, via C. Pisacane, 2 I-84100
Salerno, Italy.
ltateo, @unisa.it

Abstract. Requirements for educational software could be based on an analysis of existing learning situations. In order to obtain useful information about teaching practices, an explorative study has been conducted with a group of Italian teachers. Collected answers highlight that pedagogical support is needed in order to design effective educational software.

1 Introduction

During the design process of educational software it is very important to take into account teachers perceptions and needs [3, 16]. A software system to support face-to-face problem solving will be designed, within the LEAD project, taking into account suggestions coming from teachers in Participatory Design perspective. In order to obtain useful information about teaching practices, an explorative study has been conducted with a group of Italian teachers. A semi-structured interview was prepared in order to gather teachers' representation of computer use in classroom and, more specifically, as support for *Collaborative Problem Solving*. Collected answers highlight that pedagogical support is needed in order to design effective software supporting *Collaborative Problem Solving* into the classroom.

This research has been carried out for EU project "LEAD Technology-enhanced learning and problem-solving discussions: Networked learning environments in the classroom", funded by the VI Framework program, priority "IST Integrating and Strengthening the European Research Area".

2 Theoretical framework

This study, focused on *Collaborative Problem Solving* (CPS), belongs to the framework of cultural psychology, in particular to the collaborative and dialogical models of learning [5, 6]. Collaboration is universally recognised as a fundamental factor for cognitive development because it allows different points of view to emerge

and to be compared [14]. In such framework, CPS has been defined a complex learning process during which learners and teachers represent, analyse and try to solve different types of problematic questions. During such process collaboration and argumentation are the basis upon which new knowledge can be created [1, 2, 11, 19]. ICT tools can provide an effective support to CPS and many pieces of software have been developed, especially in educational settings, for participants interacting at a distance. In this educational paradigm [8, 18], learning mainly occurs through virtual interaction between groups of learners supported by teachers, tutors, and experts. According to [9], such educational virtual environments can be included in two main categories:

- *Action oriented systems*: environments based on the *learning by doing* principle, where learners can do actions, manipulating objects and discuss the outcomes like virtual scientific experiment software.
- *Textual production systems*: such environments are based on the collaborative production of written texts. These systems refer to the educational model that considers knowledge as built through collaboration and with the *scaffolding* of a teacher or a tutor.

Nevertheless, some studies have found that in European educational systems the most part of didactic activities are still face-to-face [7, 17]. Looking at some specific contexts, such as the South of Italy, we see many constraints to the introduction of CPS software for the everyday educational practices. For example, lack of computers into the classroom, low level of ICT expertise, and low knowledge of CPS methodology among teachers.

The Participatory Design (PD) methodology [12, 15] seems suitable to overcome at least immaterial constraints, facilitating the implementation of computer use in didactics into Italian educational context.

PD allows final user to participate to the entire process of development of technologies. In this sense, it can give the possibility to develop an environment that takes into account idiomatic and idiosyncratic communication styles of work groups. Analysis of communication technologies used in the workplaces and the analysis of users' mental models can enable designers to understand what kind of tools should be included into the software and how to display them in order to obtain the maximum advantage. Using PD, software can be designed based on teachers' needs and representations of their practices. PD in this case is useful to mediate between teachers and students knowledge mental models to build distributed systems able to give user the perception of a good affordance. Through an interface based on everyday classroom experience, the process of appropriation of new didactic support should be facilitated [13].

3 corpus, research design and categories

LEAD Italian research group collected 20 teachers' interviews from kindergarten, primary and secondary schools inquiring, among other things, what teachers know

about Collaborative Problem Solving (CPS), how they use it in classroom, and how they imagine a software system that supports face-to-face CPS.

Interview was structured in the following 4 sections composed by one or more questions: professional identity, computer use, classroom activities, and school culture. For this study, only computer use and classroom activities sections have been analysed. Interviews have been audio-recorded, transcribed considering both verbal and non-verbal aspects, and later analysed using content analysis methodology [4]. First, corpus has been read and a system of categories has been created according to research objectives and textual occurrences [10]. Categories of analysis have been attributed to the text by 5 independent judges, discussing doubtful cases until reaching a 100% level of agreement. Finally, categories' frequencies have been calculated and an interpretative analysis of teachers' answer has been conducted, on the basis of content analysis results, to reach a deeper understanding of their representation of PS and software requirements.

Interviewed teachers come from different towns of South Italy (Salerno, Avellino, Bari and Barletta). They are mainly females (19 over 20); most of them teach humanities (15 over 20) and only a few teaches scientific subjects (5 over 20). This data actually mirrors the gender distribution among Italian educational context according to the Italian Ministry of Education.

Teachers' answers have been categorized with respect to the following principal themes:

- Level of ICT expertise¹
 1. Non-users: using PC only at basic level or don't use at all;
 2. Medium-users: using PC for writing and browsing Internet;
 3. Expert-users: teachers using PC for writing and didactic.
- Definition of problem solving²
 1. Absent answer: teacher is unable to give a definition;
 2. Not pertinent answer: teacher gives a definition containing no reference to CPS model (e.g. *"I don't plan my school daily activities (...) I don't mind following the subject order (...) we decide together witch topic to discuss"*);
 3. Generic answer: teacher gives a definition containing a single reference to CPS model (e.g. *"stimulating in the student the desire to solve a situation or a question in order to obtain an answer to the problem"*);
 4. Pertinent answer: teacher gives a definition containing two or more references to CPS model (e.g. *"PS is the typical way of research questioning each situation (...) and find solution together"*).
- Examples of problem solving³
 1. Absent answer: teacher is unable to give an example;

¹ For this category the following questions of the interview have been clusterised: *"What is the role of computer in planning and carrying out your classroom activities?"*; *"What is the role of Internet in planning and carrying out your classroom activities?"*; *"How do your students use computer in classroom activities?"*; *"What is the added value of computer in classroom activities?"*.

² For this category the following question of the interview has been taken into account: *"Could you please give a definition of Problem Solving?"*.

³ For this category the following question of the interview has been taken into account: *"Could you please provide an example of using Problem Solving in your classroom?"*.

2. Not pertinent answer: teacher gives an example containing no reference to CPS model (e.g. “*studying the child emotions, so we made a circle time*”);
 3. Generic answer: teacher gives an example containing a single reference to CPS model (e.g. “*we start form a problematic situation to find the answers on each topic*”);
 4. Pertinent answer: teacher gives an example containing two or more references to CPS model (e.g. “*thinking that the plant needs water for growing (...) from their answer (...) we verify it in practice*”).
- Requirements of the CPS software⁴
1. Absent answer: teacher is unable to give a requirement;
 2. Interface/functionalities: teachers focus on the requirements concerning the software interface (i.e. with respect to users’ age) and the functionalities they think useful to support educational practices (e.g. “*perhaps with some music*”, “*helping to build diagrams*”, “*related to children’s age*”);
 3. Type of problem: teachers focus on the type of CPS the software should support and the cognitive activities involved (e.g. “*not subject related, able to manage a problematic situation*”, “*I imagine it like a real situation, I mean to ask the pupil a question he can understand*”).

4 Results

4.1 Level of ICT expertise

Teachers interviewed have a medium level of ICT expertise (Fig. 1) and only 15% (3 over 20) uses PC systematically at school, 60 % (12 over 20) uses it once per week and 25% (5 over 20) never uses PC in didactics.

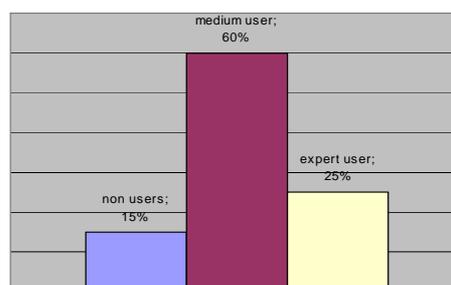


Fig 1. Teachers' expertise level

⁴ For this category the following question of the interview has been taken into account: “*How do you imagine a software supporting CPS? What characteristics would you like it to have?*”. In this case categories are not exclusive so the answer could focus on more than one aspect.

The most frequent motivation to use computer use is to be updated about their work and to prepare documents (Table. 1):

Table 1. Motivation for personal computer use.

	Italian	English
Interview #16	prendo il materiale perché io lavoro molto con la civiltà, con la cultura, e il testo che abbiamo attualmente non è molto ricco, quindi attraverso internet sperimento molta roba	<i>I get material because I work with humanities and culture, and the textbook we use is not very rich, so I experiment many stuffs through Internet</i>
Interview #10	lo uso per fare la programmazione, apportare le modifiche, stampare	<i>I use it to prepare the subject plan, to modify it, to print it</i>

Only 25% of teachers declare that they use computer also to sustain activities in classroom (Table 2):

Table 2. Motivation for computer use in didactic.

	Italian	English
Interview #17	Comunque io diciamo li porto lo stesso, <u>sia</u> per la matematica, per spiegare il computer come è fatto, e <u>sia</u> anche per esempio per musica	<i>I often bring the pupils to the lab, for mathematics, to explain the computer, also for example music</i>

4.2 Definition and examples of PS

On of the aspects emerging from the analysis of the interviews is the distance between the definition of the PS that can be found in literature⁵ and the definition provided by the teachers (Fig. 2). Only 10% of them is able to give a pertinent definition of PS, while 50% can define it only in generic terms. 25% gives a not pertinent definition.

A more problematic situation emerges when teachers are asked to give examples of PS in their didactic practices. Teachers seem unable to give practical examples coherent with theoretical definition, or with the definition they gave (Fig. 2). This gap is probably due to a superficial knowledge of PS or to the incapacity to recognize practices they use in classrooms as *Collaborative Problem Solving*.

⁵ For this study, we didn't refer to a particular definition of PS. The aim was to understand if the teachers could establish a relationship between theoretical knowledge and educational practices rather than verifying teachers qualification.

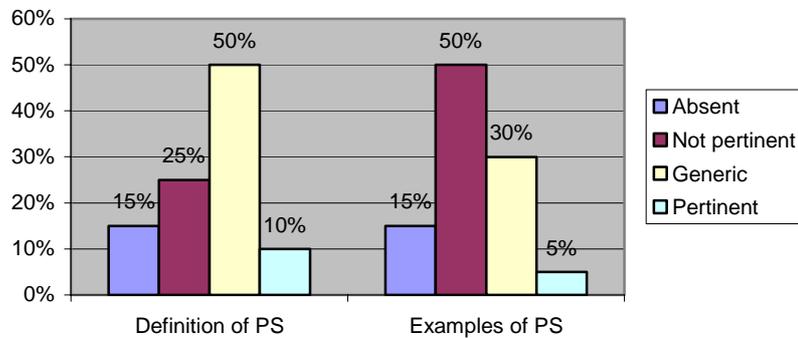


Fig. 2. Definitions of PS and examples of PS

4.3 Software requirements

The third aspect that has been investigated is requirements teachers consider relevant in a software system they would actually use in classroom. A large part of the interviewed (40%) is unable to describe any kind of features. When teachers provide an answer, it is about the interface and functionalities (Fig. 3).

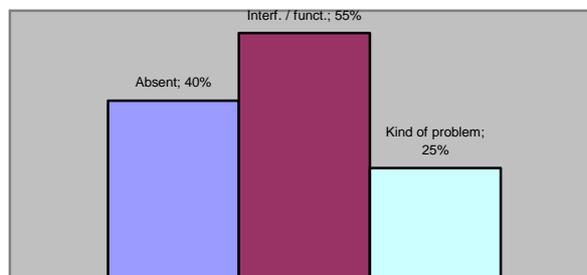


Fig. 3. Teachers' requirements for CPS software.

About interface, teachers ask for a software system that can be used autonomously by students and that is situated in children's real life (Table 3):

Table 3. Interface and functionalities requirements

	Italian	English
Interview #02	che preveda un momento in cui il bambino può utilizzarlo autonomamente ma che poi dia all'insegnante la possibilità poter concludere	<i>provide a moment in which the child can use it autonomously but then gives the teacher the possibility to finalise the activity</i>
Interview #10	i loro problemi reali, quindi mi immagino una cosa, una ripresa video e poi un'analisi successiva.	<i>about their real problems, so I imagine something with video clips and a successive analysis</i>

About type of PS, teachers imagine the software to support different problems also related to students' age (Table 4):

Table 4: Type of PS requirements

	Italian	English
Interview #12	in cui devi arrivare a una meta, risolvendo alcuni quiz.. dei giochi virtuali in cui c'è la ricostruzione di un percorso	<i>where you must reach a goal by solving some quiz. Some virtual games where you can reconstruct the path</i>
Interview #20	lo immaginerei con più soluzioni, magari lo stesso problema (...) .con gradi di difficoltà diversi	<i>with different solutions to the same problem (...) with different levels of difficulty</i>

5 Suggestions

Based on the results of these explorative interviews we can argue that:

1. teachers don't know much about CPS and they don't easily integrate it in classroom's curricular activities (e.g. "I gather the questions from children, if a pupil asks a question on his subject I find very difficult to connect it to an argument to a didactic unit that I have in my mind").
2. teachers have some requirements about a software to support Collaborative Problem Solving

Thus, design process of CPS software, at least for the Italian version, should include some kind of pedagogical support. Such support should be a short, synthetic, clear text containing theoretical, methodological, and bibliographical information.

To help teacher in designing and integrating CPS in everyday classroom activities, software should also provide a sort of template or wizard. Depending on some variables (e.g. number of students in classroom; age of students; subject of teaching), software should guide teacher through a set of phases and alternatives (e.g. problem definition; gathering information; hypotheses definition and assessment; solution of the problem). For each phase, software should present a set of possible

activities (e.g. brainstorming; meta-cognitive scaffolding; laboratory activity; role play; jigsaw; etc.)

About teacher suggestion of CPS software, we may look more coherently into the *interface/functionalities* dimension that gather the most part of teacher answers. Nevertheless, it seems clear that teachers' answers don't mention any requirement concerning interactivity and group work dimensions.

We also suggest that further investigations should be carried out in different cultural contexts to provide more information useful for the design process.

References

1. Baker, M.J., Quignard, M., Lund, K. & Séjourné, A. (2003). Computer-Supported Collaborative Learning in the Space of Debate. In: B. Wasson, S. Ludvigsen and Hoppe, U. *Designing for Change in Networked Learning Environments*. Kluwer Academic: Dordrecht.
2. Baker, M.J. (2002). Argumentative interactions, discursive operations and learning to model in science. In P. Brna, M. Baker, K. Stenning & A. Tiberghien (Eds.), *The Role of Communication in Learning to Model*, pp. 303-324. Mahwah N.J.: Lawrence Erlbaum Associates.
3. Banathy, B.H. (1992). Comprehensive systems design in education: building a design culture in education. *Educational Technology*, 22(3) 33-35.
4. Berelson (1954) "Content Analysis" in G. Lindzey, *Handbook of Social Psychology: Theory & Method*, Vol. 1, 488-522. Cambridge, Mass: Addison-Wesley.
5. Brown, A.L., & Campione, J. (1990). Communities of learning and thinking, or a context by any other name. *Contributions to Human Development*, 21, 108-126.
6. Cole, M. (1996). *Cultural Psychology*. Cambridge: Belknap.
7. Collis and Van der Wende (2002), Models of technology and change in higher education: an international comparative survey on the current and future use of ICT in higher education
8. Dillenbourg, P. (ed) (1999). *Collaborative learning: Cognitive and Computational Approaches*. Pergamon, Amsterdam.
9. Dimitracopoulou A. (2004). Designing collaborative learning systems: current trends & future research.
10. Glaser, B.G. & Strauss, A.L. (1967) *The Discovery of Grounded Theory*. Aldine: Chicago
11. Jonassen, D.H. (2000). Toward a design theory of problem solving. *Educational Technology: Research & Development*, 48 (4), 63-85.
12. Kensing, F. & Blomberg, J. (1998). Participatory design: Issues and concerns. *Computer Supported Cooperative Work*, 7, 3-4, 167-185.
13. Overdijk, M. van Diggelen, W. (2006) Technology Appropriation in Face-to-Face Collaborative Learning. Paper presented at 1th European Conference on Technology Enhanced Learning (EC-TEL 2006), 1-4 october 2006, Crete (Greece).
14. Perret Clermont, A. N. (1979). *La construction de l'intelligence dans l'interaction sociale*. Berne: Lang.
15. Perriault, J (1989). *La logique de l'usage. Essai sur les machines à communiquer*. Editions Flammarion, Paris
16. Reigeluth, C. M. (1993). Principles of educational systems design. *International Journal of Educational Research*, 19 (2), 117-131.
17. Ruthven, K. Hennessy, S. Brindley, S (2004). Teacher representations of the successful use of computer-based tools and resources in secondary-school English, mathematics and science. *Teaching and Teacher Education*, 20 (2004) 259-275

18. Slavin, R.E. (1989). Research on cooperative learning: An international perspective. *Scandinavian Journal of Educational Research*, 33 (4), 231-243
19. Resnick, L.B. Levine J.M. & Teasley S.D. (Eds.) (1991), *Perspectives on Socially Shared Cognition*. Washington DC: American Psychological Association.