2

A Conceptual and Technological Framework for Building Collaborative Learning Environments

Lonchamp Jacques
University of Nancy - LORIA
France

1. Introduction

Computer-Supported Collaborative Learning (CSCL) has emerged as a multidisciplinary research field around 1990 (Koschmann, 1996). The purpose of CSCL is to support students in learning together effectively via networked computers. During the first decade CSCL researchers have created a large number of ad hoc systems focusing at a microscopic level on particular situations and contexts and aiming at triggering specific learning processes (Lonchamp, 2006a). Their evaluation has been based mostly on the experimental paradigm, through statistical analysis of variables in relation with the learning processes, and less frequently on descriptive analysis in the ethnomethodological tradition (Stahl et al., 2006a). These early research works have contributed to the emergence of concepts, practices, and mechanisms of general interest for the CSCL domain. At the macroscopic level of institutional politics and processes, CSCL has been recognized as a possible way for preparing people to the knowledge society, for achieving deeper learning than traditional methods and for better meeting the expectations of the net generation (Resta & Lafferière, 2007). The current challenge is probably situated at an intermediary level (Jones et al., 2007), where new approaches and technologies for the effective diffusion of collaborative learning practices are needed. Many researchers think that the first generation of ad hoc, specialized, and closed tools should be replaced by systems "richer and appropriate for various collaborative settings, conditions and contexts" (Dimitracopoulou, 2005), "reconfigurable, adaptive, offering collections of affordances and flexible forms of guidance" (Suthers, 2005), "very flexible and tailorable" (Lipponen, 2002). Moreover, these generic (i.e. end-user deeply customizable) and malleable (i.e. statically and dynamically evolvable) CSCL systems should be integrated into larger environments aiming at supporting the communities of interest and communities of practice necessary for educating, training, and guiding teachers often frightened by these new approaches and technologies (Haatainen & Korhonen, 2002). The research described in this chapter explores the combination of these three concepts of "genericity", "malleability" and "community support". It is situated at an intermediary level between microscopic and macroscopic approaches, and follows the iterative design tradition. Iterative design is driven by the interactions among evolving theories, informal observations and experimentations of successive prototypes which explore the space of possible designs (Stahl et al., 2006a).

The chapter is structured into four parts. The first part outlines a conceptual framework for collaborative learning. First, the notions of collaboration and collaborative learning are discussed. Then, collaborative learning is analyzed along four dimensions: collaborative knowledge building, mediation by artifacts, distributed scaffolding and co-construction of the learning activity and its supporting system. The discussion is synthesized into a dozen of fundamental requirements. The second part describes a functional architecture that meets these requirements. This architecture includes a multi-model reflexive kernel, clients that scaffold artifact-centered collaboration and a hosting web platform for collaborative learning practice, evaluation and dissemination. The third part discusses the detailed design of the most important and original components of the proposal. Finally, the last part describes the current state of Omega+ generic synchronous CSCL system (Lonchamp, 2006a) and Escole+ hosting platform (Lonchamp, 2007c) which together implement that architecture. This proposal results from an iterative process of design and evaluation whose continuation is discussed.

2. Conceptual Framework

2.1 Collaboration and collaborative learning

There is no universally adopted meaning of the terms "collaboration" and "collaborative learning". Ingram & Hathorn (2004), define collaboration as consisting of three crucial elements: participation, interaction and synthesis. Collaboration cannot occur within a group unless there is roughly equal participation among its participants. If some participants do the main part of the work while others barely contribute, then the group is not truly collaborating. Interaction requires that group members actively respond to one another, react and change their minds as the discussion progresses. Finally, the product that the group creates must represent a synthesis of ideas and input from all members of the group. Collaboration is more than the exchange of information and ideas. It implies the synthesis of shared information and ideas that creates a product different from any that the individuals could have produced alone (Kaye, 1992), i.e. the production of new knowledge by the group. Every collaborative learning environment should at least create the basic conditions which enable collaboration: (R1) equal participation of all participants, (R2) genuine interaction among learners, and (R3) collaborative production of new knowledge.

Dillenbourg (1999) proposes four dimensions for analyzing collaborative learning: (1) the collaborative situation, including in particular the kind of artifacts that are manipulated; (2) the interactions that take place within the participants; (3) the learning processes and mechanisms; and (4) the set of effects in terms of individual and group performance. The collaborative situation is an artificial situation in which particular forms of interaction among people are expected to occur in order to trigger learning mechanisms. For increasing the probability that these forms of interaction occur, it is important to build a situation with adequate characteristics, such as:

- Group size: quite small, in most cases.
- Group composition: either homogeneous or taking advantage of a variety of knowledge and skills.
- Activity duration: rather short, in general.
- Activity orientation: for instance, an activity that stimulates critical thinking.
- Activity diversity: i.e. with unique or multiple roles.

- Activity partitioning: in general quite low; a fixed division of labor, where participants solve sub-tasks individually and then assemble the partial results into the final output, corresponds to the cooperative mode.

Interactions are characterized by their:

- Mode: rather synchronous.
- Negotiability: collaborative partners should have the possibility to negotiate how to interact at the meta-communicative level (Bateson, 1973).

Interaction rules and process structure may be imposed for facilitating productive collaboration. In a generic approach, the design of the collaborative situation, of the interaction rules, of the learning process structure, and of the way of measuring individual and group performance should be performed as far as possible by the involved teachers. This fundamental issue will be discussed in more details later.

2.2 Collaborative Knowledge Building

In several theories learning is understood through a knowledge-creation metaphor and address the same kinds of questions concerning how new knowledge is created by innovative communities (Paavola et al., 2002). Nonaka and Takeuchi's theory (1995) is based on an epistemological distinction between two sorts of knowledge, i.e. tacit and explicit. Explicit knowledge means knowledge that is easy to articulate and express formally in clear terms (Polanyi, 1962). It can be articulated in formal language including grammatical statements, mathematical expressions, specifications, manuals and so forth. Tacit knowledge means "personal knowledge embedded in individual experience and involves intangible factors such as personal belief, perspective, and the value system" (Nonaka & Takeuchi, 1995). Tacit knowledge is difficult to transfer (Spender, 1996). The dynamics of Nonaka and Takeuchi's model comes from the interaction between tacit knowledge and explicit knowledge. A knowledge spiral (see Figure 1) is based on four alternative types of knowledge conversion:

- Socialization: from tacit knowledge to tacit knowledge, when tacit knowledge and experiences are shared at the group level; socialization creates common understanding and trust within the group.
- Externalization: from tacit to explicit knowledge, when tacit knowledge is explicated and conceptualized by using concepts, models and theories.
- Combination: from explicit to explicit knowledge, when already existing explicit knowledge is exchanged, discussed, reused, and reworked.
- Internalization: from explicit to tacit knowledge, when explicit knowledge at the group level is internalized into individuals' tacit knowledge and into action. After internalization a new round in the knowledge spiral can start again.

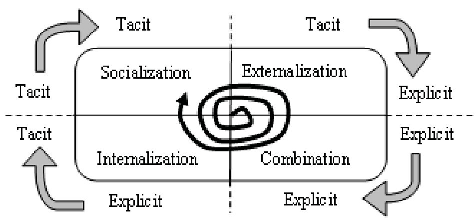


Fig. 1. Nonaka and Takeuchi's knowledge spiral

Every collaborative learning environment should (R3.1) facilitate socialization, through informal exchanges of subjectivities, emotions, opinions, doubts, (R3.2) facilitate externalization, through knowledge formalization and justification, (R3.3) facilitate combination, through knowledge comparison, synthesis, restructuring, generalizing, (R3.4) facilitate internalization, through knowledge exploration and analysis.

2.3 Mediation by Artifacts

Other theories, like Bereiter's knowledge building theory (2002), emphasize the fact that collaborating learners are elaborating, transforming, extending, and criticizing conceptual artifacts, such as ideas, methods, models and theories. Conflicting perspectives are essential to this process, through discourse activities such as questioning, proposing, arguing, critiquing, clarifying, negotiating, accusing, repairing, and agreeing. This knowledge elaboration process is greatly facilitated by knowledge materialization into shared digital artifacts (Paavola et al., 2002). These artifacts play many roles including focalization points, memories, constraints and inciters. It is worth noting that some artifacts do not contribute directly to knowledge building but serve for negotiating frames of reference such as glossaries, taxonomies, domain models, and ontologies. Figure 2 summarizes Miao's conceptual model of collaborative learning based on artifact mediation (Miao, 2000).

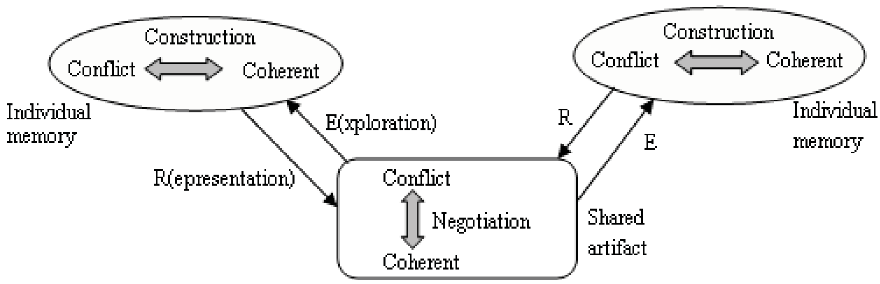


Fig. 2. Miao's conceptual model

"Representation" and "exploration" information flows can be related to Nonaka and Takeuchi's externalization and internalization processes. Knowledge in individuals' minds and the information which is held in the shared artifact can each be defined as being in "conflict" or "coherent" state. The term "conflict" has a large meaning, corresponding to all situations that can trigger a reaction from learners, including inconsistency, incompleteness, contradiction, and impreciseness. At the individual level, each learner constructs new knowledge by integrating new information into his own cognitive structure. Conflicts occur when new information contradicts existing knowledge. The learner must therefore solve this cognitive dissonance (Festinger, 1957) by constructing a new cognitive structure, i.e. by learning. At the group level, conflicts occur when one or more learners disagree with existing information in the shared workspace. In this case, the learners in the group might negotiate together to construct a new consensual state for the conflicting artifact (e.g. by combination). This dialogue can trigger learning mechanisms. Miao derives three requirements from this conceptual model of collaborative learning which make more precise the requirements deriving from Nonaka and Takeuchi's analysis. Every collaborative learning environment should (R3.2) facilitate representation of shared knowledge, (R.3.3) facilitate negotiation of shared knowledge, and (R3.4) facilitate exploration of shared knowledge.

Shared digital artifacts have physical properties that facilitate some usages, ways of thinking and objectives. Suthers (2003) has experimentally analyzed how shared, learner-constructed representations can influence collaboration in face-to-face and online situations. For instance, during an on-line collaboration with Belvedere, evidence maps are used for entering directly new ideas without preliminary discussion with the textual chat tool. Textual exchanges are used for brief confirmation dialogues and, in exceptional or problematic situations, at the metacognitive level. In a more recent work, Suthers (2006) show how many aspects suggested by knowledge construction theories can be observed experimentally during the manipulation of evidence maps: grounding (Clark & Brennan, 1991) by implicit uptake of the interlocutor's actions in the graph, interactions that respond to and address differences of interpretation, and transformations of representations by multiple individuals leading to a joint solution.

A generic environment should allow to define many different external representations and to adapt them to the specific situation and context. It should, in particular, allow the manipulation of multiple views, either partial ones, or described at different abstraction levels, or with different representation systems. When they are compared or transformed, these views can raise a new category conflicts that trigger corresponding learning processes. Multiplicity of artifacts and views makes deixis issues (referencing elements or actions within the shared workspace) more complex. Figure 3, extends Miao's conceptual model with the concept of multiple views. It also highlights the complementing roles of direct interpersonal communication and indirect communication through shared digital artifacts.

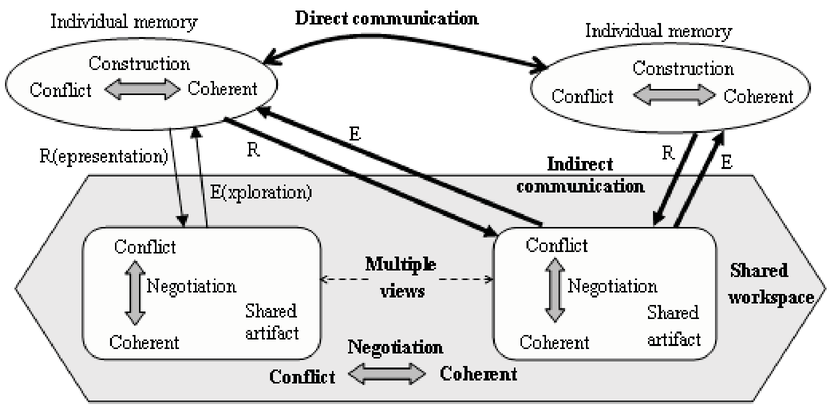


Fig. 3. The extended conceptual model

2.4 Distributed Scaffolding

The traditional concept of scaffolding involves support as provided by a teacher or more knowledgeable peer to a learner. Some researchers have tried to analyze this kind of scaffolding at a conceptual level. Bruner (1983), for instance, relates scaffolding with Vygotsky's notion of "zone of proximal development," which characterizes the region of tasks between what the learner could accomplish alone and what he or she could accomplish with assistance (Vygotsky, 1978). Bruner identifies six characteristics of effective scaffolding which are aimed at engaging and keeping the learner to task: recruitment of interest in and adherence to the task, reduction in degrees of freedom, direction maintenance, marking critical features, frustration control, and demonstration (of idealized solution paths).

A growing attention focuses on the potential of computer software to provide cognitive support for learners engaging in complex intellectual activities or “software scaffolding”. More precisely, scaffolding is distributed across the various agents that play a role in learning, including teachers, peers, software and paper and pencil tools (Puntambekar & Kolodner, 2005). Every collaborative learning environment should facilitate both (R4) software scaffolding of learners and (R5) interpersonal tutor-learner and learner-learner scaffolding. For Reiser (2002), software scaffolding may occur through two complementary mechanisms: structuring the learning task (planning, decomposition and guidance) and problematizing concepts (i.e. making something in students’ work more problematic). The software can force students to encounter important ideas or processes. While making the task more difficult in the short-term, instead of directly assisting learners, such scaffolded tools create more productive learning opportunities. Quintana’s framework (Quintana and al., 2002) proposes a set of scaffolding strategies organized around three major cognitive challenges for learners: process management (learners need support to navigate through the different work processes and activities), sense-making (learners need support to analyze and make sense of their work products to gain insight and drive the direction of their work), and articulation (learners need support to express an understanding of the work through explanations and descriptions of material they have analyzed). Table 1 summarizes the main strategies that are suggested for these cognitive challenges.

Challenge	Strategy
Process management	Structure tasks (plans, guides, visualizations…) Provide access to expert knowledge (content guidance, information for using the tool, metacognitive information) Automate non-salient, routine tasks
Sense making	Organize the tool around the semantics of the discipline (concepts and strategies) Use representations that can be inspected by learners (multiple, modifiable views) Use representations and language that bridge learners’ understanding
Articulation	Facilitate articulation of descriptions and explanations (prompting, templates…) Facilitate articulation of work plans and progress

Table 1. Quintana’s framework

2.5 Co-construction of the Learning Activity and its Supporting System

Activity Theory explains that the structure of any cooperative activity, described by Engeström (1987) in terms of object (motive), subjects, tools, rules, community and division of labor, is dynamic and continuously evolves. Rules can be bypassed, renegotiated and adjusted. Tools alter the activity and are, in turn, altered by the activity (Jonassen & Rohrer, 1999). Subjects drive these evolutions for taking into account the new needs and the contradictions that appear during the course of the activity. New activities may emerge, what Engeström (2001) refers to as “expansive learning”.

A computerized collaborative learning support system should continuously reflect the current structure of the supported learning activity, putting a strong requirement on malleability by end-users, i.e. teachers and students (Bourguin, 2000). Malleability of computerized systems is considered a difficult issue because the more efficient mechanisms are also the more difficult to use and many users are not willing to make the efforts necessary to use them. Two kinds of system malleability can be distinguished: definitional malleability, for static (i.e. before execution) adaptation to the learning situation and context as it is perceived and operational malleability, for dynamical (i.e. at run-time) flexibility. All tasks involved in designing, preparing, using, evolving and evaluating the learning activity and its computerized support constitute meta-activities in a meta-process. Bardram (1998) highlights the fact that it is a collective process, through the concept of "co-construction level" where subjects collectively reconceptualise their activity. In the specific context of collaborative learning systems, the most important meta-activities aim at:

- Designing the learning situation and customizing the CSCL system.
- Monitoring the learning process and dynamically evolving the CSCL system.
- Post-analyzing learning process results for further improvement of the situation and CSCL system.
- Supporting the technical and pedagogical development of teachers within dedicated communities of interest and communities of practice.

A community of practice consists of practitioners undertaking the same task. Learning within a community of practice takes the form of "legitimate peripheral participation" (Lave & Wenger, 1991) in which newcomers enter the community from the periphery and move toward the center as they become more and more knowledgeable. At the opposite, a community of interest brings together people interested in a given problem. Learning within a community of interest is based on "informed participation" (Arias et al., 1999) in which the knowledge is collaboratively constructed by social debate and discussion.
Every collaborative learning environment should (R6) be designed for evolution by providing a high level of malleability, both (R6.1) definitional and (R6.2) operational, together with (R7) a support for meta-activities and related interest and practice communities.

3. Architectural Design

In the light of the conceptual framework introduced in the previous section, the objective of the research can be defined as the design and implementation of a system that facilitates the creation and usage of malleable learning environments oriented towards scaffolded collaborative knowledge building through direct interpersonal communication and indirect communication via shared digital artifacts. This section discusses the main architectural design choices aiming at meeting the set of requirements summarized in table 2.

	Requirement
R1	Equal participation of all participants
R2	Genuine interaction among learners
R3	Collaborative knowledge building through:

R3.1 R3.2 R3.3 R3.4	- Socialization (informal exchanges of subjectivities, emotions, opinions, doubts...). - Externalization (knowledge formalization and justification, with simultaneous manipulation of complementary views). - Combination (knowledge comparison, synthesis, restructuring, generalizing…). - Internalization (knowledge exploration and analysis).
R4	Software scaffolding of learners (process management, sense making, articulation)
R5	Interpersonal tutor-learner and learner-learner scaffolding
R6 R6.1 R6.2	Designed for evolution by providing a high level of malleability, both: - Definitional - Operational
R7	Support for meta-activities (designing, using, monitoring, post-analyzing, educating) and related interest and practice communities

Table 2. System requirements list

3.1 A Multi-model Reflective Kernel Dealing with Exceptions

As explained in section 2.5, a malleable system (R6) can be changed by its end-users, even during its execution (operational malleability – R6.2). Mörch (1988) describes three implementation approaches of malleability, by customization, integration, and extension. Customization means to modify the parameters of existing components. Integration means to create new subassemblies of components. Extension means to change the implementation through a (more or less abstract) language. In malleability by customization, users have to select one or several values in a predefined set in order to adapt a system to their needs. System designers must anticipate all possible adaptations which is not realistic in most cases. In malleability by integration, users can add predefined components in a system in a more or less transparent way (for instance, like plug-ins installation into browsers). By creating new components it is possible to satisfy emerging needs. But the effort for developing such components is often substantial. The most direct way to achieve the conformance between the system and the supported activity is to provide a reflective system, i.e. a system which includes an explicit representation (model) of the activity. The behavior of such a reflective system depends on that (continuously queried) representation and changes when the representation is modified, thanks to the causal relationship which is implemented between the activity model and the system behavior (Maes, 1987). Omega+ generic collaborative learning system which is presented in this chapter implements this architecture extended with parameterized customization for all adaptations that can be anticipated.

Modeling cooperative learning activities both for human and machine interpretation is one of the main challenges in this approach. In the broader e-learning field, IMS Learning Design multi-level meta-model has been criticized both for its complexity and for its incompleteness, in particular for dealing with synchronous collaborative activities (Hernandez et al., 2004). The solution explored in Omega+ associates a separate (sub-) model for each facet of collaborative learning activities (in accordance with Dillenbourg's analysis evoked in section 2.1): process model, interaction model, artifact meta-model, and "effects model". This multi-model approach makes possible to build the activity representation at different levels of abstraction, adapted to the skills and needs of different categories of users by:

- Just reusing existing models.
- Building new combinations with existing sub-models (i.e. following a very high level configuration process).
- Defining or customizing simple sub-models through high-level visual languages.
- Developing complex sub-models through low-level specification and programming languages.

Omega+ reflective kernel also includes predefined tools (chat, shared text editor, shared whiteboard, and generic shared diagram editor) and mechanisms (floor control, referencing, monitoring, and group awareness) that can be selected and parameterized for fine-grained contextual adaptation.

Lastly, the kernel can also handle exceptional situations. For instance, if a learner cannot take the floor during a phase including a “round-robin” interaction protocol, a menu item allows the room operator to skip to the next learner in the circle without changing the interaction model itself. A large number of constraints can be dynamically relaxed or sidestepped thanks to ad hoc mechanisms. The system is in charge of making other users aware of these rule breakings. Table 3 summarizes these different levels of malleability.

Action	Type	Author	Objective
Model change in the library	Definitional	Teacher (as designer)	Enrichment or correction
Model change during instantiation	Definitional	Teacher (as designer)	Static adaptation
Model change during execution	Operational	Teacher (as tutor) or student	Dynamic adaptation
Constraint relaxation with ad hoc mechanisms	Operational	Teacher (as tutor) or student	Exceptional events handling
Parameter change	Operational	Teacher (as tutor) or student	Personalization

Table 3. Different levels of malleability

3.2 A Dual Interaction Space with Software Scaffolds

Omega+ provides on the client side a “dual interaction space” including a communication space and a task space (Dillenbourg et al., 2005). Such a dual space allows collaborative knowledge construction through direct interpersonal exchanges in the communication space, and indirect communication in the task space, via artifacts sharing. Several research works demonstrate that it is not easy for learners to use efficiently such dual spaces for cognitive and meta-cognitive exchanges, even in the simplest configuration with a chat and a whiteboard (Dillenbourg & Traum, 1999). Various scaffolding mechanisms are required. In Omega+, they mainly rely on the four sub-models evoked in the preceding section. They are discussed in relation with requirements R1 to R5.

3.2.1. Equal participation of all participants (R1)

Direct communication can be constrained by protocols either predefined or ad hoc, i.e. specified via interaction models. For instance, equality of participation can be imposed by the "round robin" predefined protocol (Fuks et al., 2006). Ad hoc protocols including application-related roles such as "presenter," "criticizer," "reviewer," "synthesizer," constitute another way of regulating participation (Pfister & Mülpfordt, 2002), in particular when roles are exchanged between participants. Moreover, in a dual interaction space, protocols can be used to control both direct and indirect communication if the "right to speak" includes the "right to manipulate" shared artifacts (Lonchamp, 2007a).

In a less constraining manner, meta-cognitive tools which visualize participation characteristics can also incite participants to perform best, either directly through self-regulation or indirectly through a tutor (Jermann, 2004). In Omega+, "effects models" are used for producing customized meta-cognitive tools (Lonchamp, 2008).

3.2.2. Genuine interaction among learners (R2)

Dialog structuring with tailorable sentence openers (Soller, 2001) is a first elementary technique provided by Omega+ for promoting effective interaction through questioning, clarification, agreement, contradiction, etc.

Typed messages and application-related roles specified into interaction models constitute a more powerful technique for promoting particular forms of interaction. For instance, the reviewer role can be associated with specific interaction types such as "correct," "comment" or "complement" (O'Donnel & Dansereau, 1992).

Some artifacts, corresponding to explicit representations of the space of debate between learners (Baker et al., 2003), can play a similar structuring role for indirect communication.

Lastly, referencing means (between tools and between spaces) can make easier questioning and reacting (Lonchamp, 2007b).

3.2.3. Collaborative knowledge building (R3)

Socialization (R3.1), i.e. informal exchanges of subjectivities, emotions, opinions, doubts, etc., is not easy when participants are not in a face to face setting. The interest of adding a video/audio channel to the textual chat is complex to analyze. A study from (Scholl et al., 2006), for instance, demonstrates no effect on dialog regulation and effects on task content highly dependent on video quality. Chat and whiteboard will likely remain the preferred tools for informal exchanges in distributed settings.

A large spectrum of artifact types makes easier externalization (R3.2). This spectrum ranges from loosely structured artifacts, such as concept graphs or representations using the card metaphor (Cox & Greenberg, 2000), to formal artifacts with a precise operational semantics allowing to animate them, like finite state automata or Petri nets. All these artifact types, independent of their degree of formality, are defined through artifact models using ontologies adapted to the domain and to the learners. During artifact construction, prompting techniques can help to trigger and guide properties elicitation.

Composition (R3.3) requires simultaneous manipulation of several artifacts of the same type or of different types in the same shared workspace. Omega+ workspace can be configured with multiple tools specified in the process model. Knowledge exploration can be made easier thanks to various kinds of representations and mechanisms such as hierarchical representations (multi-level graphs), multi-page representations (in whiteboard tools), thumbnail views for manipulating complex artifacts and a session history search window (Lonchamp, 2009).

3.2.4. Software scaffolding of learners (R4)

Besides sense making and articulation, scaffolding also concerns process management. This point meets the collaborative learning scripting research stream. Scripts aim at formalizing the sequence of activities, the way the task is distributed within the group and the mode of interaction among the participants (Jermann & Dillenbourg, 2003). Recent works distinguish between micro-scripts which specify the activities of individual learners during collaborative phases and macro-scripts which specify the overall structure of the learning process (Dillenbourg & Tchounikine, 2007). For instance, some micro-scripts based on role switching describe application-related roles and the way learners are statically or dynamically assigned to these roles (Pfister & Mülpfordt, 2002). Macro-scripts emphasize the idea that collaborative activities are embedded into larger processes including also individual and cooperative activities. It is the case for instance when learners write personal positions before a collaborative debate on a given topic. In the following, we use the terms "micro-process model" and "macro-process model," instead of micro and macro scripts. These models can either inform participants of the process they should follow (Carell et al., 2005) or guide them step-by-step through their interpretation by the supporting system (Wessner et al., 1999). In this last approach, implemented in Omega+, it is important to avoid procedural over-specification (Dillenbourg, 2002) and straight-jacket effects (Schmidt & Bannon, 1992). A high level of operational malleability is required when (micro or macro) process models are machine-enforced.

3.2.5. Interpersonal scaffolding (R5)

This kind of scaffolding relies in part on aspects that are not directly related to the supporting environment, like group composition: for instance, mixing different levels of experience and competence can favour peer scaffolding (Lai & Law, 2006).

Some functionalities of direct communication tools can also help interpersonal scaffolding between learners or between tutors and learners. For instance, chat tools should provide private channels for more individualized help than through the public channel. Lastly, all the approaches for genuine interaction described in section 3.2.2., also contribute to interpersonal scaffolding.

3.3 A Web Platform for Collaborative Learning Design, Practice, Evaluation and Dissemination

Collaborative learning tools slowly disseminate in real settings. Two important reasons are the lack of technical and pedagogical expertise from regular teachers and the lack of concrete assistance and guidance they can receive. The solution which is explored in this research is a web platform, called Escole+, aiming at satisfying several objectives (R4, R7):

- Support the collective learning macro-processes, including both collaborative activities supported with Omega+ and cooperative or individual activities.
- Support the collective meta activities of modelling, practicing and evaluating collective learning macro-processes.

- Host the communities of practice around these learning processes and meta activities.
- Host the global community of interest including teachers simply curious, or interested by, or practicing collaborative learning and specialized researchers.

All participants, including students, can access Escole+ platform and Omega+ environment by using a regular web browser.

This approach generalizes the communities of practice for the professional development of teachers, such as Tapped In 2 (Schank et al., 2002), by adding to an information-oriented support a process-oriented support. The "Community, Content and Collaboration Management System" concept (C3MS) is also closely related to Escole+. A C3MS is an extension of a Content Management System (CMS), like Zope or PostNuke, with bricks for constructing socio-constructivist scenarios: communication and argumentation tools (synchronous and asynchronous), project-based learning tools, tools for designing scenarios, etc. (Schneider et al., 2002). The C3MS approach suffers from serious flaws resulting from the weaknesses of CMS as integrating kernels. For instance, CMS are specialized for web content production processes and cannot support arbitrary defined macro-processes.

3.4 Functional Architecture

The functional architecture reflecting the previous design choices is summarized in Figure 4. The next section discusses the detailed design of its more important and original components.

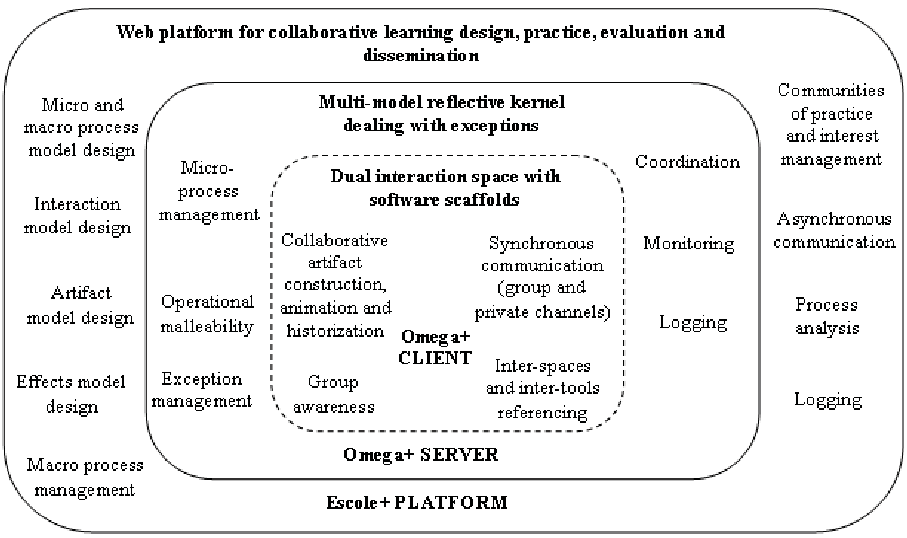


Fig. 4. The proposed functional architecture

4. Detailed Design

The following subsections describe the detailed design of the most important and original components of the proposal. Figure 5 summarizes the overall conceptual model of Omega+ and helps for understanding how the different sub-models are structured and interrelated.

It is worth noting that all important concept types - roles, tools, protocols, floor control policies (FCPs) - are specialized into predefined and application-specific sub types. This reflects the fact that Omega+ provides both hard-coded mechanisms and model-based features which are customizable and evolvable by its users.

4.1 Collaborative Micro-Process Modeling

Activity theory highlights the importance of plans for guiding work in collaborative settings (Bardram, 1997). A plan is not a rigid prescription of the work to be done but a guide that can be changed in accordance with the context during its execution. In Omega+ a collaborative micro-process specifies a sequence of phases taking place into rooms, i.e. shared spaces: simple phases where all the participants collaborate to the same activity into the same room and split phases where participants perform simultaneously different activities within sub groups into distinct rooms which all start and terminate synchronously. A plan “A;B;C” does not prescribe the execution of phases A, B and C exactly in that order. ABBC, AB, AB’C (where B’ is a modified version of phase B), ABCBC are other possible execution traces. Concretely, participants playing the predefined role of “Room operator” have two buttons for selecting the next phase to execute, either by following the plan (“Next” button) or by selecting another existing phase in the plan (“Jump” button), i.e. starting a “process deviation” which is explained below.

A micro-process model includes a set of phase types, a set of tool types, precedence links between phase types and inclusion links between phase types and tool types. Each phase

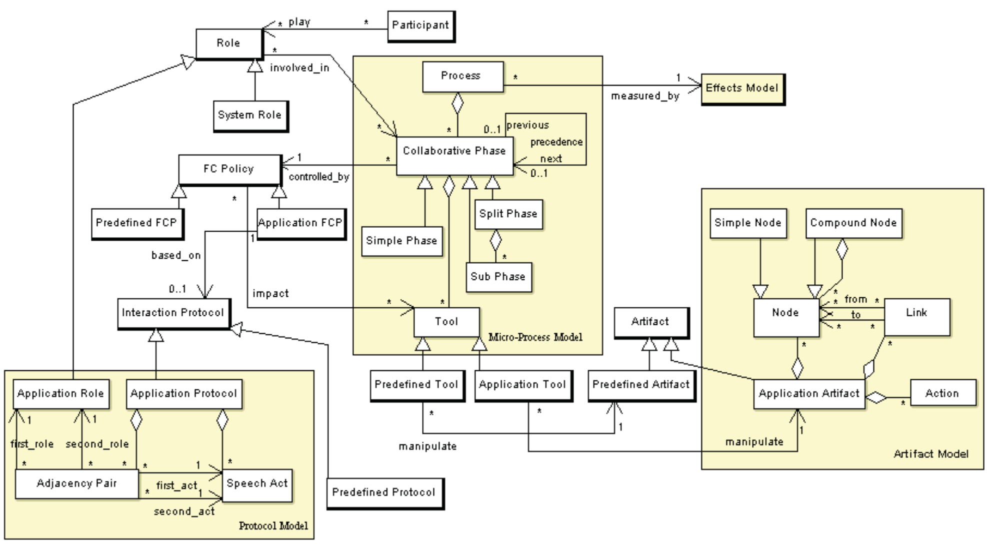


Fig. 5. Omega+ conceptual model

type is characterized by its name, its type (simple or split), its informal description, its interaction model (see section 4.2), the coordination mode (see section 4.5) which applies into the room(s) and the phase types which cannot be reached from that point with the “Jump” button. Each tool type is characterized by its name, its type (shared text editor, shared whiteboard, shared artifact editor), a boolean indicating if the tool is in read-only mode and, possibly, the name of a file that should be automatically loaded into the tool at the beginning of the phase (or the name of its type) and the name of a file that should be automatically created by the tool at the end of the phase. A micro-process model can be created off line on Escole+ platform, with a shared graphical editor or directly in its XML representation. A micro-process model can also be created interactively by a Room Operator when it is launched. The same forms are also used for changing dynamically the micro-process model definition. When a phase instance is created, the Room Operator:

- Gives it a name (by default the type name plus the instance number).
- Defines its participants, if room access is controlled, and the binding of participants to application-related roles defined in the interaction model (for instance, who is the Moderator in a phase with a moderated interaction protocol).
- Can provide informal instructions to the participants.
- Can parameter the chat tool with sentence openers (when it is consistent with the interaction model), sentence numbering and explicit referencing through these numbers.

When a process deviation is requested, the phase to which control should be transferred (or "END" for terminating the process) is selected. This process deviation request is analyzed on the basis of the deviation tolerance information included into the process model. If the requested phase belongs to the explicit list of exclusions associated to each phase type the deviation is refused and a message is displayed explaining that this deviation would violate the intrinsic constraints of the process model. In the other case, the jump is accepted and a "deviation step" takes place before the effective start of the selected target phase. During this step, all input and output resources are displayed and can be freely edited by the tutor. This includes input resources declared as read-only in the process model. Output resources are opened with their last content which is automatically saved each time a phase terminates. The system also helps in determining which phases and resources are potentially impacted ("polluted") by the deviation process.

4.2 Interaction Protocol Modeling

Interaction protocol models define specific patterns of textual communication. The set of predefined protocol models (such as "round robin," "moderated," "unique contributor") can be extended by application-dependent models. These ad hoc models define a set of application-dependent roles, a set of application-dependent message types (speech acts) and a set of adjacency pairs (Clark & Schaefer, 1989) saying "if a participant playing the role X send a message of type M1 then a participant playing the role Y – anyone, the next one in a circular ordering, the same one if X=Y - must send a message of type M2". The protocol model also defines which role(s) can speak first. An interaction model can be created off line on Escole+ platform, with a shared graphical editor or directly in its XML representation. Dynamic change of the protocol model definition is not supported. However, it is possible to change at any moment the model which applies during a phase and some exceptional situations are managed for predefined protocols, such as skipping a participant in a round robin phase or temporary excluding a participant.

4.3 Artifact Modeling– the Generic Shared Artifact Editor

An artifact model, which is in fact a meta-model, formalizes how the generic shared artifact editor is customized for manipulating target models of a given type. These target models are defined as hierarchical graphs in which a node can be refined into a sub-graph. Among the predefined meta-models one can find for instance finite automata, UML schemas, concept graphs, Petri nets, IBIS graphs (Kunz & Rittel, 1970), conceptual graphs (Sowa, 1984), logical circuits and card-based approaches (Cox & Greenberg, 2000). Teachers can also define any ad hoc representation by specifying three aspects. (1) The visual representation of node types (icon for the editor button, icon into the graph, label position, property list with, for each property, its name, type and prompt) and edge types (icon for the editor button, colour, dash-dot pattern, arrow type, type-related label, instance-related label). (2) The structural constraints defining which node types can be related by which edge types, through a structural graph. (3) The operational semantics possibly attached to node types and to the overall graph. The specification of actions is made by program like in CoolModes (Pinkwart, 2003). An action name can be attached to a node type. The action name appears on the contextual menu of nodes of this type and triggers a local action. The local action is specified through a method with the same name in the class associated to the node type. This class inherits from a predefined class in the kernel called DiagramNode which allows accessing all useful information about a node. An "action" component can also be integrated into a graph. It creates a specific "action button" in the editor whose name and icon are properties of the action component. The action button triggers a global action on the graph. The global action is specified through a method with the same name in the class associated to the graph type. This class inherits from a predefined class in the kernel called DiagramFormalism which allows accessing to all useful information about a graph. All these methods work on a server-side representation of the graph and send visual modification messages to the clients when it is required (e.g. change the colour, label or icon). By this way, it is possible to animate representations such as Petri nets or logical circuits and, more generally, to associate any transformation to the artifacts and their components.

The generic shared artifact editor takes an artifact model as parameter. Besides this definitional malleability, it also provides a number of sophisticated functionalities such as browsing the different levels of hierarchical graphs, browsing their construction history (Mühlpfordt & Stahl, 2007) and browsing large graphs through a bird's-eye view allowing direct manipulation.

4.4 Effects Modeling – Meta-cognitive tools

In Omega+, the process of building high-level visual representations (meta-cognitive tools) from low level raw data is generic. The "effects model" describes how the indicators are computed and presented.

Omega+ distinguishes between (Lonchamp, 2008):

- Simple indicators which are combination of predefined low-level variables presented as stacked time series or stacked bar charts: they can measure for instance the participation (number of messages, number of tool actions) which should be roughly equal for all learners, the balance between conversation events and action events for verifying that both spaces are used, the communication style (message size) because rich and explicit messages are necessary for externalizing knowledge or the usage of the referencing mechanisms.
- Complex indicators that are based on complex customizable mechanisms. A first example is a mechanism for distinguishing between "on-task" and "off-task" messages. The underlying algorithm is a Naïve Bayes Classifier extended with stemming and

stop-words removal. The specialized language to be learned by the classifier is defined through a set of resources including the files loaded into the text board (e.g. the problem description), the meta-model files which parameter the generic diagram editor (for learning concept type names), the created diagrams (for learning node and link names) and a set of explicit textual resources specified into the "Effects model" (e.g. formalism descriptions). A second example of complex customizable mechanism is a mechanism for counting patterns reflecting either interaction situations (for instance, two participants who modify successively the same component or directly related components in a graph) or actions aiming at facilitating interaction (for instance, a participant who modifies a graph and immediately sends an on-task message with the chat tool). The "Effects model" contains explicit pattern definitions that use a simple but extensible pattern definition language.

4.5 Coordination Mechanism

The dual interaction space provides both a multi tools task space and a communication space where interaction can be regulated by protocols. This raises several issues in terms of user coordination:

- Granularity level where coordination policies should apply: environment, space, artifact, component.
- Possible coexistence of several different policies.
- Relationships between communication and coordination of actions.

Omega+ provides a small number of global floor coordination policies (FCPs) at the environment level. These global policies are specified, for each phase, in the micro-process model. Every global policy is defined in terms of local policies at the space level. Table 4 summarizes all the possibilities which are detailed and illustrated in (Lonchamp, 2007a). "Free floor" means that users have no restriction for accessing the tools in the space without consistency guarantee (unless if the tool itself ensure some consistency at a finer grain level). "Exclusive control" is implemented through a button for asking the floor, a button for releasing the floor and an adaptable floor assignment policy. By default, it is a FIFO policy (first in first out). It is also possible to define a delay time before preemption. The Room Operator can also explicitly pass the floor from a participant to another. Regulation through a protocol ("protocol-based policy"), either predefined or ad hoc, can impact only the communication space or the whole environment.

Global policy	Task space policy	Communication space policy
Free floor	Free floor	Free floor
Free talking, exclusive doing	Exclusive control (adaptable)	Free floor
Free doing, exclusive talking	Free floor	Exclusive control (adaptable) or protocol-based
Parallel floors	Exclusive control (adaptable)	Exclusive control (adaptable) or protocol-based
Common floor	Exclusive control (adaptable) or protocol-based	

Table 4. Omega+ coordination policies

Thanks to this set of policies, all the options for floor acquisition defined by Myers (2000) – explicit, by designation, by protocol-, unless implicit acquisition which is problematic in a multi tool context, and all options for floor release – explicit return, explicit taking, preemption on inactivity- are made available.

4.6 Referencing Mechanism

Linking conversations and task objects is fundamental for establishing comprehension and shared attention (Stahl et al., 2006b). There are many different techniques to achieve referencing. Textual messages can include spatial references (“the blue square on the right”), temporal references (“your last created node”), quotes (“in the message where you say …”), references to participant’s name or pseudo and references to line numbers. Graphical editors can provide non persistent mechanisms such as telepointers (Hayne et al., 1994), mechanisms for changing objects appearance when they are designated with the mouse (Suther et al., 2003) and graphical pointers that shade progressively (Dongqiu & Gross, 1999). Graphical editors can also provide persistent mechanisms like graphical pointers and annotations, either textual (Fidas et al., 2001) or graphical (Giordano & Mineo, 2005), whose positioning implicitly designate the referenced object. Finally, at the environment level, Concert-Chat provides explicit inter tool links (Mühlpfordt & Wessner, 2005).
The more complex links, with several sources and/or several targets, which are likely to occur into multi tools dual spaces, are difficult to express by these means. Omega+ provides a new application-independent mechanism which generalizes annotations and allows complex referencing. In all tools of both spaces users can stick freely annotated snapshots of the whole environment (or SAS for “sticky annotated snapshot”). SAS can be included into other SAS, thus creating powerful discussion threads. Textual annotations and persistent pointers are available as simplified versions of SAS. Besides the classical usage for referencing (for instance, showing the correspondences between two graphs or highlighting the main ideas in a set of chat messages through circling, connecting and underlining) some more surprising usages have been observed (Lonchamp, 2007b). Firstly, SAS can serve as fully fledged intermediary objects as defined by (Vinck & Jeantet, 1995), i.e. ephemeral and shared representations appearing during collaborative design processes which serve as mediators to discussion and reflect some transformation or translation of the designed artifact. Secondly, SAS can help to bypass constraints which result from some coordination policies. They can be used either as a private space when free floor policy applies, because SAS are not shared before they are saved, or as a way to communicate when the exclusive control policy applies and the floor is away, because SAS creation is not submitted to any control unlike all other communication channels.

4.7 Modeling Environment

Process models (micro and macro), interaction models and artifact models take the form of graphs with specific properties attached to their components. The modeling environment is simply a collaborative space where the generic shared artifact editor is parameterized with the corresponding meta-models and extended by generation actions for producing the internal XML representation of models.
Other tools can enrich this modeling environment like a concept map editor for discussing the knowledge content of the envisioned learning activity and editors of high level notations for pedagogical design (Lonchamp, 2008).

4.8 Escole+ Platform

Escole+ web platform includes three specialized spaces:
(1) A pedagogical space, including:

- A community space for exchanging general information among the community of interest through asynchronous communication tools such as forums, wikis or document sharing areas.
- A design space where Omega+ models and macro-process models are developed by teachers and specialists within specialized subspaces for each project, through the modeling environment described in the previous section and asynchronous communication tools for the communities of practice.

(2) A learning space, where tutors and students execute model-driven learning processes within subspaces dedicated to each process instance.
(3) A platform administration space, for managing Escole+ users, generic roles, backups, etc.
Macro-process management raises some specific issues. Few collaborative platforms provide workflow support and when it exists the level of operational malleability they provide is low. In a first experiment (Escole platform) this kind of support has been evaluated (Lonchamp, 2006b). It was hard to implement and rigid, for the small benefit of automating some repetitive tasks. In Escole+, we take advantage of the possibility of dynamically creating hierarchical workspaces, assigning them resources and controlling their access for different categories of users (Lonchamp, 2007c). Controlling the process is done simply by dynamically modifying the access rights to the workspaces where the different activities defined in the macro-process model (individual, cooperative or collaborative) are implemented. These workspaces can be generated from the macro-process models which specify the activities, tools, roles, precedence and inclusion relationships.

5. Implementation and Evaluation

Omega+ is written in Java and can run either as a client/server application on a LAN with socket communications or as a web application (applets for clients and servlet for the server) with HTTP tunnelling for dealing with firewalls. Escole+ is build on top of Libresource cooperative platform (www.libresource.org) developed in the same research team. Quite all the functionalities summarized in Figure 4 are implemented. Only, the post mortem analysis tool and the tool for replaying off line collaborative sessions from the log file are still under development.
A series of preliminary experiments have been conducted with Omega+. For instance, a collaborative micro-process model for object oriented design has been tested with computer science students. In this model, small groups of four students receive the description of a situation in the read-only text editor that can be seen in the middle left pane in Figure 6. Their objective is to build a UML class diagram corresponding to that situation. During the first phase of the two-phase process whose structure is displayed in the "Model structure" sub-window at the bottom of Figure 6, students must specify a set of use cases through short textual descriptions (with the shared editor in the top left pane) and draw the use case diagram (with a specialization of the generic shared graph editor in the bottom left pane). The global policy "free talking, exclusive doing" is associated with this phase for allowing a free debate like during a brainstorming but with a strict coordination for accessing the shared artifacts. Figure 6 shows Jack's client who owns the floor at this moment (see the

"release floor" button on the top left and the white backgrounds of editors that indicate the right to contribute – unless for the read only editor). All participants, including Jack, can communicate at any moment through the chat tool of the communication space on the right. As Jack also plays the generic role of Room Operator, he can use the "Next" and "Jump" buttons, described in section 4.1, for flexibly controlling the micro-process.

The second phase is the core of the design activity. Students can see the use cases previously defined in the read only text editor in the top left pane of Figure 7. They transform these use cases into collaboration diagrams, with the customized shared graphical editor in the middle left pane, and introduce progressively new classes into the class diagram, with the customized shared graphical editor in the bottom left pane. For guaranteeing both a disciplined way of working and equality of participation, the interaction model called "CircularWork" is used for piloting the environment. Each student can take the floor in turn for acting with the editors and commenting with the chat tool. The predefined "round robin" protocol is not well adapted because the floor changes of owner after each contribution. The "CircularWork" protocol is a model-defined protocol that allows making several contributions before passing explicitly the floor to the next learner in the circle.

Figure 8 shows the "CircularWork" model in the shared graph editor of the modeling environment. At each moment, a learner can only select with the combo list at the bottom of Figure 7 one specific type of message in accordance with his/her role and the adjacency pairs in the interaction model ("Say" or "Pass" for the floor owner and nothing for all others). In the chat history the sequence of messages reflects this protocol. It is worth noting that at every moment the Room Operator could change this global policy and select for instance a "free floor" policy.

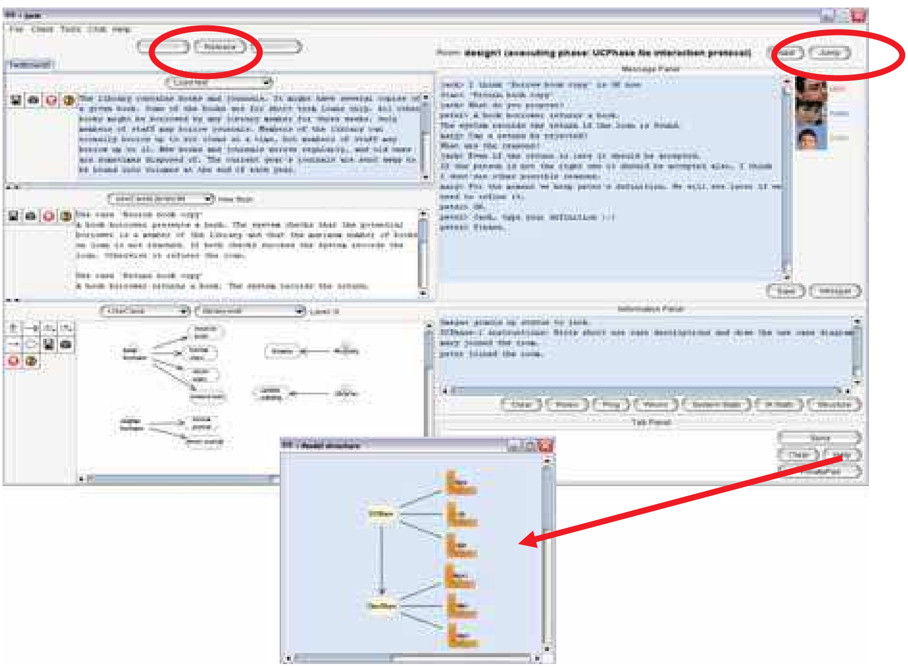


Fig. 6. Jack's client during the first phase

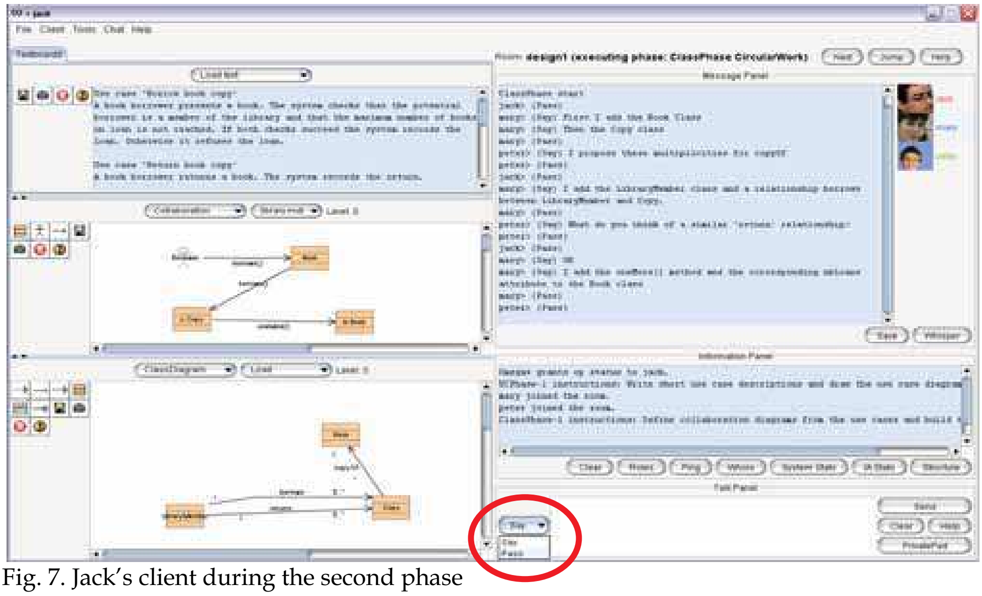


Fig. 7. Jack's client during the second phase

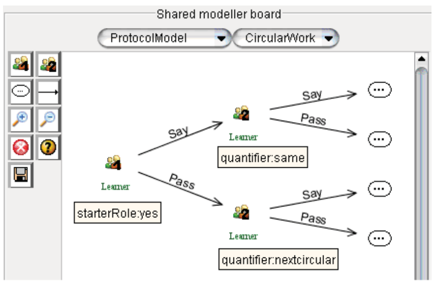
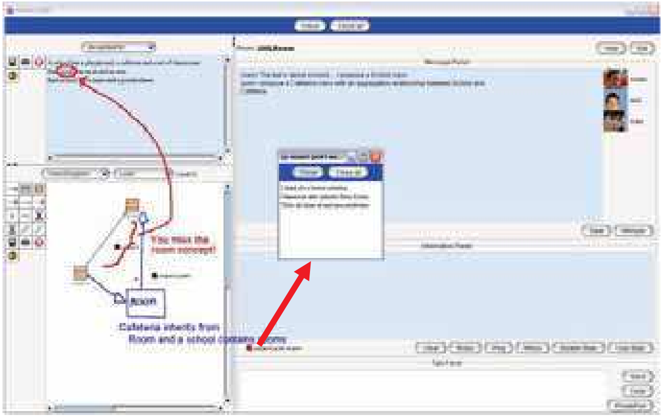
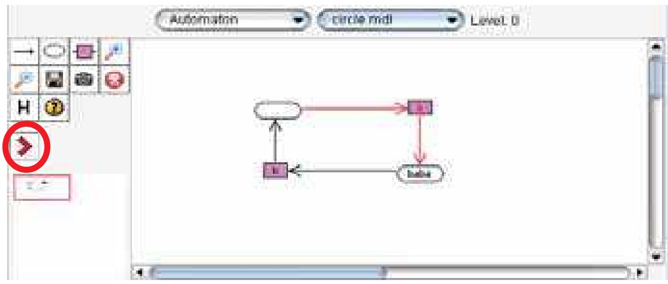
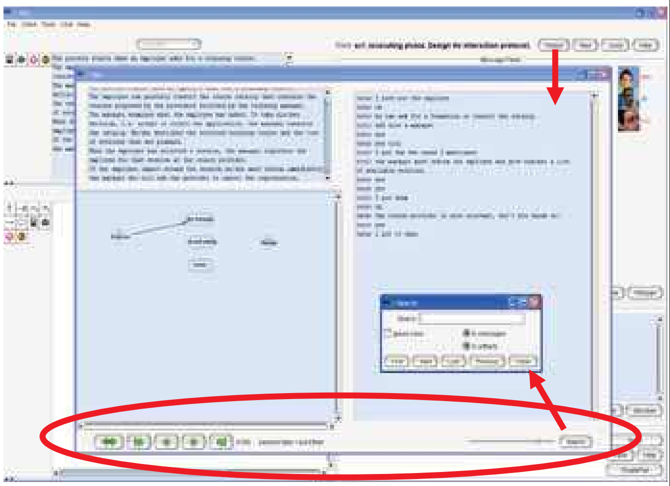


Fig. 8. The "CircularWork" protocol model

This example focuses on collaboration guidance through micro process models, coordination policies and protocol models. Additional examples of mechanisms supporting learners at the cognitive and meta-cognitive levels are shown in Table 5.

Due to the number of its functionalities and its malleability the environment is difficult to evaluate. All classical evaluation approaches evoked in the introduction, either quantitative or qualitative, can only evaluate design elements by isolating them. This problem has already been highlighted by other researchers: "As Co-Lab is a large comprehensive system, evaluation studies have had to focus on specific aspects of it, rather than evaluating the whole system" (van Joolingen et al., 2005).

Level–Objective–Description	Example
Cognitive–Referencing- A SAS displayed in the SAS viewer. It contains free hand drawings, annotations and an example of inter-tool reference. It contains several other SAS. One of them is a sticky note which has been opened in its own viewer.	
Cognitive -Artifact animation and scrollable bird's-eye view- This graphical editor contains an action button (in the middle left part) for animating the finite state automaton and a bird's-eye view for quick browsing (in the bottom left part).	
Cognitive–Session history browser- Participants can browse the history step by step by directional buttons or with a slider located in the bottom panel of the browser. When a learner presses the "sync" button on the left of the bottom panel a browser is automatically launched in each client environment (if it is not already started) and all browsers are synchronized for enforcing a shared focus on a given point of the process history. A textual search facility (on the right) is also provided.	

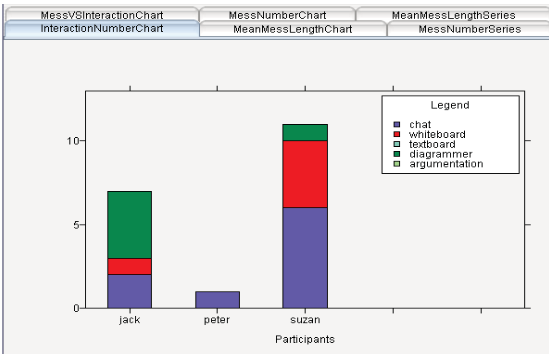
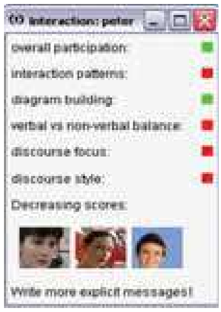
Meta cognitive– Customizable high level visualizations- A stacked histogram of the number of initiatives for each learner and each tool.	
Meta cognitive-Interaction analysis- A customizable interaction report: for each criterion a red (green) square means a result under (over) the average result. Learners are also classified by decreasing aggregated scores. A guidance message is also displayed for the worse indicator (Lonchamp, 2008).	

Table 5. Cognitive and meta-cognitive support

The central question in our approach is the capacity of teachers and learning technology providers to make use of the definitional and operational malleability provided by Omega+. A first step in that direction has been done by defining a methodological approach for conducting qualitative interaction analysis oriented toward the improvement of the supporting environment that can be applied to any learning task and any environment configuration. This “generic analysis approach” is organized into three levels (Lonchamp, 2009):

(1) At the dialog level, a task-independent dialogical model is proposed for analyzing action/communication traces as “generalized conversations”. A graphical notation is provided for visualizing the syntactical characteristics of collaborative sessions.

(2) At the knowledge level, a typology of task-independent collaborative knowledge building episode types that can occur during such generalized conversations is proposed. Thanks to that classification scheme, recurrent meaningful elements that structure the low-level descriptions can be detected and characterized. These regularities help for passing from local interpretations to a global interpretation of the whole process.

(3) At the action level, task-dependent socio-cognitive interpretations of why the collaborative learning process unfolds as observed are proposed. These interpretations can constitute a firm basis for improving the customization of the generic environment in order to support learners more efficiently.

In the next future, we plan to conduct both cross-application comparisons in search of differences, commonalities and generalisations at the macro level, and cross-configuration comparisons for the same application in search of the best supporting strategies and mechanisms at the micro level, leading to the further iterative improvement of both the technical infrastructure and the methodological approach.

6. References

Arias, E. G.; Eden, H.; Fischer, G.; Gorman, A. & Scharff, E. (1999). Beyond access: Informed participation and empowerment, *Proceedings of Int. Conf. on Computer Supported Collaborative Learning*, pp. 20–32, Stanford, California, Lawrence Erlbaum Associates, Mahwah, N.J.

Baker, M.J.; Quignard, M.; Lund, K. & Séjourné, A. (2003). Computer-supported collaborative learning in the space of debate, *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, pp. 11-20, Bergen, Norway, Kluwer Academic Publisher, Dordrecht.

Bardram, J. (1997). Plans as Situated Action: an Activity Theory Approach to Workflow Systems, *Proceedings European Conf. on Computer Supported Cooperative Work*, pp. 17-32, Lancaster, U.K., Kluwer Academic Publisher, Dordrecht.

Bardram, J. (1998). Designing for the dynamics of cooperative work activities, *Proc. Int. Conf. on Computer Supported Cooperative Work*, pp. 89-98, Seattle, Washington, ACM Press.

Bateson, G. (1973). *Steps to an Ecology of Mind*, Granada, London, U.K.

Bereiter, C. (2002). *Education and Mind in the Knowledge Age*, Lawrence Erlbaum Associates, Hillsdale, N.J.

Bourguin, G. (2000). Un support informatique à l'activité coopérative fondé sur la Théorie de l'Activité : le projet DARE, *PHD Thesis, University of Lille*, France.

Bruner, J. (1983). *Child talk: Learning to use language*, Norton, New York.

Carell, A.; Herrmann, T.; Kienle, A. & Menold, N. (2005) Improving the Coordination of Collaborative Learning with Process Models, *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, pp.18-27, Taipei, Taiwan, ISLS.

Clark, H. H. & and Brennan, S. A. (1991). *Grounding in communication. Perspectives on socially shared cognition*, APA Books, Washington.

Clark, H. & Schaefer, E. (1989). Contributing to Discourse. *Cognitive Science*, Vol. 13, 259-294.

Cox, D. & Greenberg, S. (2000). Supporting collaborative interpretation in distributed Groupware. *Proceedings Int. Conf. on Computer Supported Cooperative Work*, pp. 289-298, Philadelphia, Pennsylvania, ACM Press.

Dillenbourg P. (1999). What do you mean by collaborative learning? In: *Collaborative-learning: Cognitive and Computational Approaches*, 1-19, Elsevier, Oxford, U.K.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design, In: *Three worlds of CSCL. Can we support CSCL*, P.A. Kirschner (Ed.), 61-91, Open Universiteit Nederland, Herlen.

Dillenbourg, P. & CSCL SIG of Kaleidoscope (2005). Dual Interaction Spaces. *CSCL'05 workshops presentation* http://www.cscl2005.org/Workshops/workshop5.htm.

Dillenbourg, P. & Tchounikine, P. (2007). Flexibility in macro-scripts for computer-supported collaborative learning, *Journal of Computer Assisted Learning*, Vol 23, 1–13.

Dillenbourg, P. & Traum, D. (1999). Does a shared screen make a shared solution? *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, pp. 127-135, Stanford, California, Lawrence Erlbaum Associates, Mahwah, N.J.

Dimitracopoulou, A. (2005). Designing Collaborative Learning Systems: Current Trends & Future Research Agenda, *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, pp. 115-124, Taipei, Taiwan, ISLS.

Dongqiu, Q. & Gross, M.D. (1999). Collaborative Design with NetDraw, *Proceedings Int. Conf. on Computer Aided Architectural Design Futures*, pp. 213-226, Atlanta, Georgia, Kluwer Academic Publisher, Boston.

Engeström, Y. (1987). *Learning by expanding*, Orientakonsultit, Helsinki.

Engeström, Y. (2001). Expansive learning at work: Towards an activity theory reconceptualisation, *Journal of Education and Work*, Vol 14, 133–156.

Festinger, L. (1957). *A Theory of Cognitive Dissonance*, Stanford Univ. Press, Stanford, CA.

Fidas, C.; Komis, V. & Avouris, N. (2001). Design of collaboration-support tools for group problem solving, *Proceedings Panhellenic Conf. on Human Computer Interaction*, pp. 263-268, Patras, Greece, Typorama Publications, Patras, Greece.

Fuks, H.; Pimentel, M. & Pereira de Lucena, C. J. (2006). R-U-Typing-2-Me? Evolving a chat tool to increase understanding in learning activities. *ijcscl*, Vol 1 (1), 117-142.

Giordano, D. & Mineo, S. (2005). A graphical annotation platform for Web-based e-learning. *Proceedings Int. Conf. on Multimedia and Information and Communication Technologies in Education*, pp. 1255-1260, Cáceres, Spain, FORMATEX, Badajoz, Spain.

Haatainen, E. & Korhonen, K. (2002). Guidelines for teacher training and technical and pedagogical support. ITCOLE teacher training and consulting model, *ITCOLE Project Deliverable D8.1*, IST-2000-26249, http://www.euro-cscl.org/site/itcole/D8_1_guidelines_ for_teach.pdf.

Hayne, S.; Pendergast, M. & Greenberg, S. (1994). Implementing Gesturing with Cursors in Group Support Systems. *Journal of Management Information Systems*, Vol. 10, 42–61.

Hernández-Leo, D.; Asensio-Pérez, J. I. & Dimitriadis, Y. (2004). IMS Learning Design support for the formalization of Collaborative Learning Patterns, *Proceedings Int. Conf. on Advanced Learning Technologies*, Joensuu, Finland, pp. 350-354, IEEE Press.

Ingram, A.L. & Hathorn, L.G. (2004). Methods for Analyzing Collaboration in Online Communications, In: *Online collaborative learning: theory and practice* (T.S. Roberts, ed.), 215-241, Idea Group Inc, Hershey.

Jermann, P. & Dillenbourg, P. (2003). Elaborating new arguments through a CSCL scenario, In: *Arguing to Learn: Confronting Cognitions in Computer – Supported Collaborative Learning Environments*, 205–226, CSCL Series, Kluwer, Amsterdam, Holland.

Jermann, P. (2004). Computer Support for Interaction Regulation in Collaborative Problem-Solving, *Doctoral Dissertation*, University of Geneva.

Jonassen, D.H. & Rohrer-Murphy, L. (1999). Activity Theory as a Framework for Designing Constructivist Learning Environments, *Educational Technology, Research and Development*, Vol. 47 (1), 61-79.

Jones, C.; Dirckinck-Holmfeld, L. & Lindström, B. (2007). A Relational, Indirect, Meso-Level Approach to CSCL Design in the Next Decade, *ijcscl*, Vol. 1 (1), 35-56.

Kaye, A. (1992). Learning together apart, In: *Collaborative Learning Through Computer Conferencing*, A.R. Kaye (Ed.), 117-136, Springer-Verlag, Berlin, Germany.

Koschmann, T. (1996). *CSCL: Theory and Practice of an Emerging Paradigm*, Lawrence Erlbaum Associates, Hillsdale, N.J.

Kunz, W. & Rittel, H. (1970). Issues as Elements of Information Systems. *Technical Report S-78-2. Insitut für Gundlagen Der Planung I.A., Universität Stuttgart*, Germany.

Lai, M. & Law, N. (2006). Peer Scaffolding of Knowledge Building Through Collaborative Groups with Differential Learning Experiences, *Journal of Educational Computing Research*, Vol. 35 (2), 123-144.

Lave, J. & Wenger, E. (1991). *Situated learning-legitimate peripheral participation*. Cambridge University Press, New York.

Lipponen, L. (2002). Exploring foundations for computer-supported collaborative learning. *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, Boulder, Colorado, pp. 72-81, Lawrence Erlbaum Associates, Hillsdale, N.J.

Lonchamp, J. (2006) Supporting synchronous collaborative learning: A generic, multi-dimensional model, *ijcscl,* Vol. 1 (2), 247-276.

Lonchamp, J. (2006). A Platform for CSCL Practice and Dissemination, *Proceedings Int. Conf. on Advanced Learning Technologies*, Kerkrade, Holland, pp. 66-70, IEEE Press.

Lonchamp, J. (2007). Floor Control in Complex Synchronous CSCL Systems. *Proc. Int. Conf. on Web Information Systems and Technology*, Barcelona, Spain, pp. 397-402, INSTICC.

Lonchamp, J. (2007). Linking Conversation and Task Objects in Complex Synchronous CSCL environments, *Proceedings Int. Conf. on Web Information Systems and Technology*, Barcelona, Spain, pp 281-288, INSTICC.

Lonchamp, J. (2007). Towards a Web Platform for Collaborative Learning Practice, Evaluation and Dissemination, *Journal of Computers*, Vol. 2 (4), 1-8.

Lonchamp, J. (2008). Interaction Analysis Supporting Participants' Self-regulation in a Generic CSCL System, *Proceedings Third European Conference on Technology Enhanced Learning*, Maastricht, Holland, pp 262-273, LNCS 5192, Springer.

Lonchamp, J. (2008). Designing Collaborative Learning Applications, *Proceedings 8th IEEE International Conference on Advanced Learning Technologies*, pp. 353-355, Santander, Spain, IEEE Press.

Lonchamp, J. (2009). A three-level analysis of collaborative learning in dual interaction spaces, ijcscl, Vol 4(3), 289-317.

Maes, P. (1987). Concepts and experiments in computational reflection. *Proceedings Int. Conf. on Object-oriented programming systems, languages and applications*, Orlando, Florida, pp. 147–155, ACM Press.

Miao, Y. (2000) Design and Implementation of a Collaborative Virtual Problem-Based Learning Environment, *Ph.D. Thesis, Technischen Universität Darmstadt*.

Mørch, A. (1995). Three Levels of End-user Tailoring: Customization, Integration, and Extension, *Proceedings Third Decennial Aarhus Conference*, pp. 157-166, Aarhus, Denmark, Department of CS, Aarhus University.

Mühlpfordt, M. & Wessner, M. (2005). Explicit Referencing in Chat Supports Collaborative Learning, *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, pp. 460-469, Taipei, Taiwan, ISLS.

Mühlpfordt, M. & Stahl, G. (2007). The integration of synchronous communication across dual interaction spaces, *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, pp. 525-534, Rutgers, New Jersey, ISLS.

Myers, B.A.; Chuang, Y.S.A.; Tjandra, M.; Chen, M.C. & Lee, C.K. (2000). Floor Control in a Highly Collaborative Co-Located Task, *Tech. Report - Peebles Project*, http://www.cs.cmu.edu/~pebbles/papers/pebblesfloorcontrol.pdf.

Nonaka, I. & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*, Oxford University Press, New York.

O'Donnell, A. & Dansereau, D. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance, In: *Interaction in cooperative groups - the theoretical anatomy of group learning*, 120-141, Cambridge University Press, New York.

Paavola, S.; Lipponen, L. & Hakkarainen, K. (2002). Epistemological foundations for CSCL: A comparison of three models of innovative knowledge communities, *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, Boulder, Colorado, pp. 24-32, Lawrence Erlbaum Associates, Hillsdale, N.J.

Pfister, H.-R. & Mühlpfordt, M. (2002). Supporting discourse in a synchronous learning environment: The learning protocol approach. Proceedings Int. Conf. on Computer Supported Collaborative Learning, Boulder, Colorado, pp. 581-589, Lawrence Erlbaum Associates, Hillsdale, N.J.

Pinkwart, N. (2003). A Plug-In Architecture for Graph Based Collaborative Modeling Systems. *Proceedings Int. Conf. on Artificial Intelligence in Education*, Sydney, Australia, pp. 535-536, IOS Press, Amsterdam.

Polanyi, M. (1962). *Personal knowledge: towards a post critical philosophy*, Routledge, London.

Puntambekar, S. & Kolodner, J.L. (2005). Toward implementing distributed scaffolding: Helping students learn science from design. *Journal of Research in Science Teaching*, Vol. 42 (2), 185-217.

Quintana, C.; Reiser, B. J. ; Davis, E. A. ; Krajcik, J. ; Fretz, E. ; Duncan, R. G. et al. (2004). A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences*, Vol 13 (3), 337-386.

Reiser, B. J. (2002). Why Scaffolding Should Sometimes Make Tasks More Difficult for Learners, *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, Boulder, Colorado, pp. 255-264, Lawrence Erlbaum Associates, Hillsdale, N.J.

Resta, P. & Laferrière, T. (2007). Technology in Support of Collaborative Learning, *Educational Psychology Review*, Vol 19, 65–83.

Schank, P.; Harris, A. & Schlager, M. (2002). Painting a Landscape onto TAPPED IN 2, *Proc. CSCW Workshop on The Role of Place in Virtual Communities*, New Orleans, Louisiana, http://tappedin.org/tappedin/web/papers/2002/TI2Place CSCW.pdf

Schmidt, K. & Bannon, L. (1992). Taking CSCW Seriously. Supporting Articulation Work. *Journal of Computer Supported Cooperative Work*, Vol 1 (1-2), 7-40.

Schneider, D.; Synteta, P. & Frété, C. (2002). Community, Content and Collaboration Management Systems in Education: a new chance for socio-constructivist scenarios? *Proceedings Int. Conf. on Information and Communication Technologies in Education*, Samos, Greece, pp. 49-56, July, 2002, FORMATEX, Badajoz, Spain.

Scholl, J.; McCarthy, J. & Harr, R. (2006). A comparison of chat and audio in media rich environments, *Proceedings Int. Conf. on Computer Supported Cooperative Work*, Banff, Alberta, Canada, pp. 323-332, ACM Press.

Soller A. (2001). Supporting Social Interaction in an Intelligent Collaborative Learning System, *Journal of Artificial Intelligence in Education*, Vol. 12, 40-62.

Sowa J.F. (1984). *Conceptual Structures : Information Processing in Mind and Machine*, Addison-Wesley, Reading, Massachusetts.

Spender J.C. (1996). Making Knowledge the Basis of a Dynamic Theory of the Firm, *Strategic Management Journal*, Vol. 17 (Winter Special Issue), 45-62.

Stahl, G.; Koschmann, T. & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective, In: *Cambridge Handbook of the Learning Sciences.* Cambridge University Press, Cambridge, U.K.

Stahl, G.; Zemel, A.; Sarmiento, J.; Cakir, M.; Weimar, S.; Wessner, M. & Mühlpfordt, M. (2006). Shared Referencing of Mathematical Objects in Online Chat, *Proceedings Int. Conf. of the Learning Sciences*, Bloomington, Indiana, pp. 716-723, ISLS.

Suthers, D.; Girardeau, L. & Hundhausen, C. (2003). Deictic Roles of External Representations in Face-to-face and Online Collaboration, *Proceedings Int. Conf. on Computer Support for Collaborative Learning,* Bergen, Norway, pp. 173-182, Kluwer, Amsterdam, Holland.

Suthers, D. (2005). Technology Affordances for Intersubjective Learning: A Thematic Agenda for CSCL, *Proceedings Int. Conf. on Computer Supported Collaborative Learning*, Taipei, Taiwan, pp. 662-671, May, 2005, ISLS.

Suthers, D. (2006). A Qualitative Analysis of Collaborative Knowledge Construction Through Shared Representations, *Research and Practice in Technology Enhanced Learning*, Vol 1 (2), 1-28.

van Joolingen, W.; de Jong, T.; Lazonder, A.; Savelsbergh, E. & Manlove, S. (2005). Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning, *Computers in Human Behavior*, Vol. 21, 671–688.

Vygotsky, L. S. (1978). *Mind in Society: The development of higher psychological processes*. Harvard University Press, Cambridge, Massachusetts.

Vinck, D. & Jeantet, A. (1995). Mediating and Commissioning Objects in the Socio technical Process of Product Design: a conceptual approach, In: *Designs, Networks and Strategies*, COST Social Science Series, 2.

Wessner, M.; Pfister, H.-R. & Miao, Y. (1999). Using learning protocols to structure computer-supported cooperative learning, *Proceedings World Conf. on Educational Multimedia, Hypermedia & Telecommunications*, Seattle, Washington, pp. 471-476, AACE.

Advanced Learning

Edited by Raquel Hijn-Neira

ISBN 978-953-307-010-0
Hard cover, 444 pages
Publisher InTech
Published online 01, October, 2009
Published in print edition October, 2009

The education industry has obviously been influenced by the Internet revolution. Teaching and learning methods have changed significantly since the coming of the Web and it is very likely they will keep evolving many years to come thanks to it. A good example of this changing reality is the spectacular development of e-Learning. In a more particular way, the Web 2.0 has offered to the teaching industry a set of tools and practices that are modifying the learning systems and knowledge transmission methods. Teachers and students can use these tools in a variety of ways aimed to the general purpose of promoting collaborative work. The editor would like to thank the authors, who have committed so much effort to the publication of this work. She is sure that this volume will certainly be of great help for students, teachers and researchers. This was, at least, the main aim of the authors.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Lonchamp Jacques (2009). A Conceptual and Technological Framework for Building Collaborative Learning Environments, Advanced Learning, Raquel Hijn-Neira (Ed.), ISBN: 978-953-307-010-0, InTech, Available from: http://www.intechopen.com/books/advanced-learning/a-conceptual-and-technological-framework-for-building-collaborative-learning-environments

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.