# Towards a Web Platform for Collaborative Learning Practice, Evaluation and Dissemination

Jacques Lonchamp
LORIA, Campus Scientifique, BP 239, 54506 Vandœuvre-lès-Nancy Cedex, France
Email: jloncham@loria.fr

***Abstract*—Computer Supported Collaborative Learning (CSCL) tool adoption is slow and challenging when compared to the dissemination of simpler content delivery systems. The lack of technical and pedagogical expertise, concrete training, and effective support for teachers are some reasons that explain the current situation. This paper proposes a technological approach for making CSCL practice, evaluation and dissemination easier. In this approach, a powerful workflow-enabled web platform hosts virtual communities of volunteer teachers and CSCL specialists. The platform supports participants for designing collective learning processes that integrate CSCL tools, executing and tutoring them, analyzing their results and debating all related technical and pedagogical issues.**

***Index Terms*—CSCL, dissemination, evaluation**

## I. INTRODUCTION

Computer Supported Collaborative Learning (CSCL) systems reflect the importance of social processes as an essential element of learning. Broad concerns about the limitations of traditional educational approaches in an increasingly global and technological world emphasize the need to realize the potentials of collaborative learning and computer support. Despite the production of an impressive number of tools and environments by the CSCL community during its first decade, CSCL adoption remains slow and challenging when compared to the dissemination of more classical e-learning environments supporting instructionalist pedagogy [1].

Early CSCL tools suffered from restricted applicability (limiting reuse), in the sense that they were characterized by very specific situations, particular forms of interaction and specific learning processes. It was the case of early specialized synchronous tools for structured discussion [2, 3], collaborative design [4, 5], knowledge construction [6], modeling [7] and writing [8]. Conversely, future technologies are described by the CSCL community as richer and appropriate for various collaborative settings, conditions and contexts (Dimitracopoulou [9]), reconfigurable, adaptive, offering collections of affordances and flexible forms of guidance (Suthers [10]), very flexible and tailorable (Lipponen [11]). A recent analysis of synchronous CSCL systems shows, during the last five years, the beginning of a trend towards systems with a larger applicability and flexibility [12]. Many recent systems use explicit models as parameters (i.e., 'model-based genericity'), for defining the kind of artifacts which are collaboratively manipulated [13, 14, 15], the interaction dimension [16, 17, 18], and the process dimension [19]. All these approaches have in common an increased complexity for teachers resulting from the existence of non trivial collaborative 'meta-activities' for: (1) designing the learning situation (model design) and customizing the CSCL system (model instantiation), (2) monitoring the learning process and adapting on the fly the supporting system (dynamic model evolution), (3) post-analyzing learning process results for further improving the CSCL system (static model evolution), (4) supporting the pedagogical development of teachers within a community of practitioners. Three of these cooperative activities (1, 3, and 4) are clearly outside the scope of purely synchronous CSCL systems.

This paper proposes a powerful workflow-enabled web platform for hosting virtual communities of volunteer teachers and CSCL specialists. This platform should provide support for designing collective learning processes that integrate CSCL tools, executing and tutoring them, analyzing their results and debating all related technical and pedagogical issues. All participants, including students, should access the system through a standard web browser. The use of such a platform can be justified on the grounds of the benefits brought about by learning in a web-based environment, such as flexibility as regards time and place, and support for knowledge building. It is also in line with the idea of encouraging teachers to create networks to share their expertise.

The next section defines the main platform requirements from the end-user perspective and the corresponding logical architecture. The third section describes ESCOLE, a first prototype of J2EE-based implementation, through a typical usage scenario. Section four describes a second prototype, ESCOLE+, aiming at improving the approach and supporting more complex learning activities. Finally, section 5 compares the proposal with related works.

Based on "A Platform for CSCL Practice and Dissemination", by J. Lonchamp which appeared in the Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, 2006, Kerkrade, The Netherlands, July 2006. 

## II. Requirements And Logical Architecture

This section defines the basic platform requirements, mainly from the end-user perspective, and the corresponding logical architecture.

(R1) The platform should support both teachers' and students' learning. The pedagogical development of teachers can take different forms. Newcomers can learn about pedagogical, technical and practical issues directly, by observing ongoing processes (sessions), in a similar way of what is described in open source communities [20]. They can learn also indirectly, by reading reports and best practices catalogs, and by communicating with CSCL specialists and other teachers. Later, observers can start to participate to collective learning activity definition and design. Finally, they can tutor activities, possibly with the help of more experienced teachers.

(R2) The platform should provide a set of specialized shared workspaces. Teachers and CSCL specialists share a 'pedagogical space' in which they perform all activities for preparing, evaluating and discussing CSCL projects. This pedagogical space should include a hierarchy of project-based sub spaces. A 'production space' is set-up by the organizers of each process in which students perform collectively the prescribed CSCL activities, generally under the control of tutors. When necessary, the production space is split into specialized sub spaces for different classes or groups. Additionally, all participants (teachers and students) can perform individual work in private spaces on their personal computers.

(R3) The platform should provide basic cooperative tools and CSCL tools, selectively accessible from the different workspaces. The basic cooperative toolset includes production tools (e.g., shared editors, document sharing facility), communication tools (e.g., email, news, forums, chat, wiki), and coordination tools (e.g., shared calendar, vote facility). Of course, in their private spaces, all participants can continue to use their preferred production tools. In addition, learning sessions make use of different kinds of synchronous CSCL tools, either dedicated to a specific task or generic, i.e. parameterized by explicit models of learning processes. Unlike basic cooperative tools, CSCL tools rely on well-identified learning theories, such as creating Piagetian socio-cognitive conflicts or participating in social processes of knowledge production in line with Vygotsky's ideas.

(R4) The platform should provide active assistance and guidance for both teachers and students. The diversity of roles and situations makes impossible to constrain learning activities to follow some predefined pedagogical project structure [21], or some generic cooperation pattern, like the 'Researching-Confronting-Structuring' pattern [22]. Assistance and guidance should rely, at least in part, on the interpretation of workflow models by a full-fledged workflow service integrated into the platform. The same idea can be found in virtual campus platforms, mainly for course management [23]. The kind of processes we have in mind is slightly different: synchronous CSCL sessions are often integrated within larger collective learning processes including loosely coupled cooperative tasks where students and tutors simply share documents. For instance, a process for developing analysis and synthesis abilities can start with an evaluation of the initial skills of each student (e.g., by individually writing a content summary of a given document). The second step is the CSCL session, using a synchronous argumentation tool for collaboratively learning about the same document. The process terminates with a new individual summary production for evaluating the benefit of collective argumentation for each student. The platform must support such hybrid processes, mixing individual and collaborative activities, synchronous and asynchronous work. In addition, collective activities for preparing learning processes often include routine management tasks which can also be workflow-enabled. In this approach, control and guidance stay at two levels. At the global level, workflow models describe the activity flow, roles and resources of collective learning processes, roughly speaking at the same granularity level than IMS Learning Design specifications (http://www.imsglobal.org/learningdesign). Local process and protocol descriptions, describing at a fine grain level how students and tutors interact during synchronous sessions, are either hard-coded in conventional CSCL tools, or explicitly specified in models serving as parameters for generic CSCL tools.

(R5) The platform should provide various forms of awareness information and rough monitoring data. At the global level, awareness is independent of the learning technique, and includes workspace awareness, process awareness, and shared document state awareness. At the local level, fine-grained awareness relies on the characteristics of the learning approach. Each tool (communication tool, editor, CSCL tool …) and platform service (workflow, document sharing …) should generate event traces on the server side. These traces should be customizable, in line with the observations and analysis that are required. Automatic trace analysis tools (e.g., [24]) could sometimes be used.

(R6) The platform should provide runtime malleability to end-users (tutors and students). Models for the workflow service and for generic CSCL tools are specified by their designers in accordance with their pedagogical intents and the implementation constraints. However, some level of dynamic malleability, i.e. end-user model evolution at runtime and/or exception management, is necessary for taking into account the context-dependent and situated nature of learning processes.

## III. The ESCOLE Prototype

The aim of the ESCOLE prototype ('Environment for Supporting COllective Learning Enthusiasts') is to experiment typical usage scenarios for validating the approach. For building the prototype, we have searched for existing J2EE-based cooperative work support environments, including basic communication tools, a document sharing facility and a workflow engine. The open source Coefficient platform has been selected (http://coefficient.sourceforge.net). At the application level, we have integrated ωChat, our open source generic framework for building malleable structured chat applications for distributed educational settings [25]. ωChat combines explicit process models (like [19]) and explicit protocol models (like [17]). In the next section, we describe one of our usage scenarios and its implementation with the prototype.

### *A. A Typical Usage Scenario*

The initiator of the project uses the announcement mailing list of the platform for its presentation and for calling for participation. Then, the initiator creates a sub space of the pedagogical space providing communication, document sharing, and issue tracker tools. The initiator uploads the project description, its schedule, and possibly an initial version of its 'direction documents': informal directives, Coefficient global workflow model, ωChat local models. Interested teachers and CSCL specialists are enrolled as organizers, with write permissions to the pedagogical sub space, tool access, and registration into the project definition mailing list. Other members of the community and anonymous visitors are observers, with only read permissions. The collective pedagogical engineering work produces the final version of the global and local models. It takes place in the pedagogical sub space and can use past experiment reports and a best practices catalog (similar to http://tecfa.unige.ch/proj/seed/catalog) all available in the pedagogical space.

When the engineering phase is terminated, the production space is set up with the required resources and tools. Organizers can test the new learning process on line. When tests are completed, tutors are chosen among organizers for performing actual learning sessions. Tutors receive write permissions to the production space and its direction documents, tool access, and are registered in the project performance mailing list. All other members of the community and external visitors only receive read permissions. Finally, students are registered on the platform by tutors. They can use the tools of the production space with write permissions, and are registered in the project performance mailing list. The starting date of the session is set and sent through the project list and the news.

The collective learning process defined and used during our scenario aims at studying how collaborative work can improve the skill at understanding and summarizing complex documents. The global workflow model has the following structure:

1. *Initial summarization step*: each student, of a small group of three to five, writes an initial summary of the text containing the knowledge to be acquired.

2. *CSCL step:* during a synchronous session without any active tutor or expert, students follow the COTEXT method [17]. The text is divided into as many sections as there are students. To each section is associated a two steps iteration (ωChat process model):

   *2.1.Production step:* a student, playing the Summarizer learning role, produces a summary.

   *2.2.Review step:* the other students act as Commentators in accordance with the EXP protocol depicted in Fig.1 (ωChat protocol model).

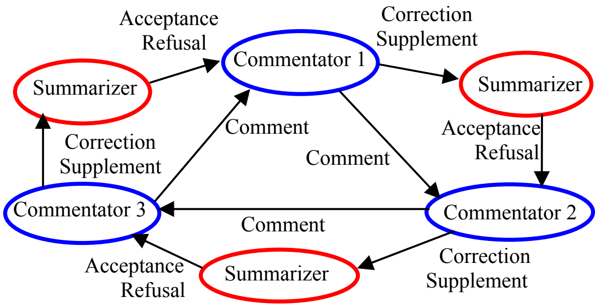


Figure 1. The EXP protocol.

At the beginning of each iteration, the Summarizer role is taken by the next student. The iteration ends when there are no more Corrections or Complements.

3. *Final summarization step:* each student produces individually a new summary. With all initial and final summaries teachers can evaluate the efficiency of the collective learning process.

### *B. The Implementation of the Scenario*

After identification, the user can see the list of all ESCOLE projects (workspaces). In Coefficient, there is no notion of sub space. Projects in the list can only be classified into categories and characterized by some attributes. Several communication tools are available from everywhere in the navigation bar. When entering a project, the specific tools of this project have icons to launch them. In the case of a project driven by a workflow model, directives for the current step are displayed for the user. Some actions can automatically trigger the transition to one of the possible next steps in accordance with the transition rules of the workflow model. Automatic actions can be executed when entering or leaving a step (entry and exit actions). Fig. 2 shows ESCOLE client when entering the 'Initial summarization step' of the scenario within the production space 'Summary1'. The user knows the current step ("You are in <step_name> of workflow <workflow_name>") and how he/she should behave ("For this stage you must: <directive_text>").

A folder 'InitialSummaries' is automatically created by an entry action (see Fig. 3). It comes with the 'Document' folder where the tutor has uploaded the document under study. As soon as one file for each student has been uploaded with the 'File uploads' tool, the workflow engine triggers the next step. In this 'CSCL step' (see Fig. 4) ωChat appears as a tool integrated within ESCOLE. Coefficient provides means for accessing the tool with an icon, displaying a descriptive tooltip and a help file, and storing produced data in 'MyData' personal page. The 'PublishLog' tool complements ωChat for moving the chat log file into the upload/download area which constitutes the shared information space of the process. This kind of tool is necessary in Coefficient because automatic actions can only be attached to the entry or exit of a step. No specific event is attached to tool invocation or closing. With the log file and the summaries, tutors can create the session report that will be automatically stored into the persistent pedagogical space, called here the 'Community' project

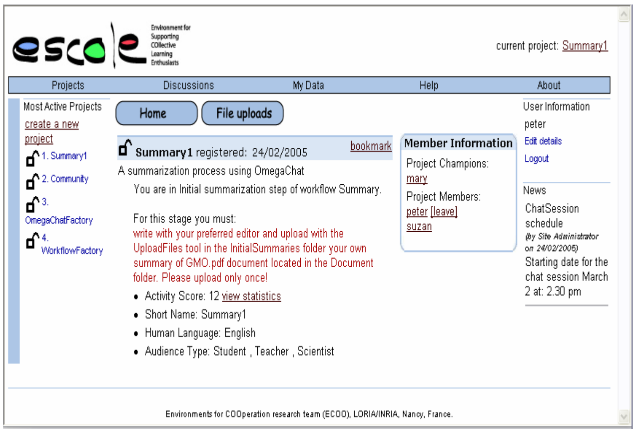


Figure 2. The 'Initial Summarization' step.

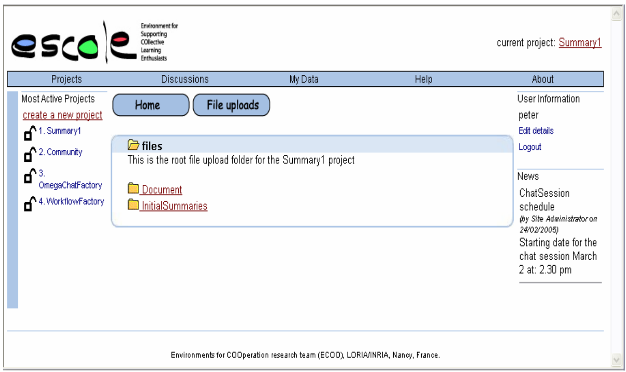


Figure 3. The generated folders.

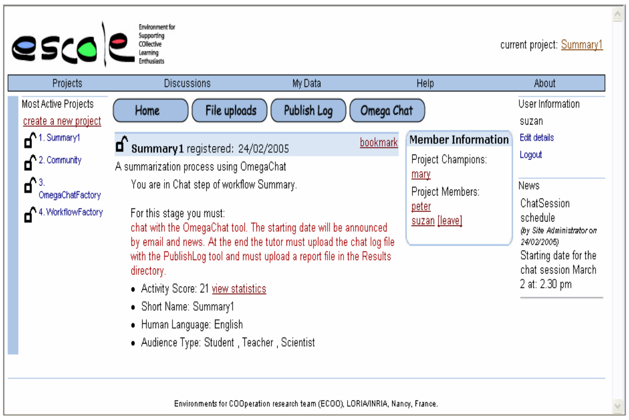


Figure 4. The CSCL Chat step.

Fig. 5 shows ωChat tool during the collaborative review (step 2.2). The EXP protocol is enforced as users can only choose between a list of protocol-permitted utterance types when they have the floor (see the 'Can you talk?' indicator on top of the 'Talk Panel' and the 'Say' combo list). Correction, Supplement, Comment utterance type labels appear at the beginning of each broadcasted message after the author's name.

The development of the workflow model has taken place within the 'Workflow Factory' project. In this permanent project, workflow models (XML files) can be uploaded and downloaded with the 'File uploads' tool. Each time a new version of a model is uploaded, the workflow model which controls the 'Workflow Factory' project, automatically compiles and deploys this uploaded

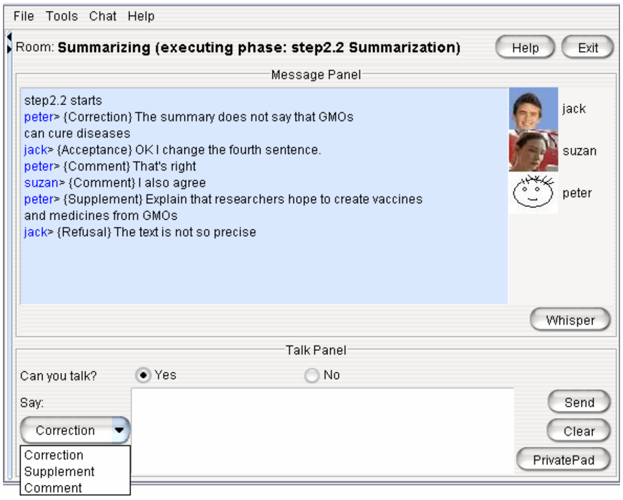


Figure 5. ωChat tool (review step).

model. When it is done, one can use this model for driving a new project. The same principle holds for the permanent 'OmegaChatFactory' project, where organizers can develop and install XML process and protocol models for ωChat. This illustrates the importance of workflow-enabled processes both for CSCL activities (preparing resources, automating transitions) and for community activities (automating tedious tasks like model compiling and deployment).

## IV. ESCOLE+: TOWARDS A PRODUCTION PLATFORM

ESCOLE prototype suffers from important limitations mainly resulting from Coefficient characteristics. For instance:

– projects (workspaces) cannot be organized in a tree structure as suggested in requirement R2,

– creating a new project implies a lot of tedious manipulations, such as selecting and configuring tools, mailing lists, folder structure, and so on; this work must be repeated for every project,

– no real event management service is available; such a service could highly improve activity monitoring, tool integration and awareness functionalities as explained in requirement R5,

– no graphical modeling facilities are available neither at the workflow level nor at ωChat level. In ESCOLE, teachers have to manipulate low level XML descriptions.

During the last year we have released simultaneously a new version of the synchronous CSCL framework, called Omega+ [12], richer and more flexible, and a new hosting platform, called ESCOLE+, solving all above-mentioned limitations and providing a best support for dealing with the increased complexity resulting from Omega+.

### A. From ωChat to Omega+

Omega+ applies 'model-based genericity' to the four dimensions of collaborative learning: the situation (manipulated artifacts), the interaction (protocol), the process, and the way of monitoring individual and group performance. These aspects are explicitly specified in four models (artifact meta-models, protocol, process and effect models) that serve as parameters for the generic environment. This opens the possibility of flexibly defining and combining many scaffolding techniques that have been proposed in isolation in the CSCL literature: process scripts, customized sentence openers, speech acts and interaction protocols, application-specific artifacts, ad-hoc meta-cognitive tools, etc. Omega+ modeling approaches are described in detail in [12]. Omega+ favors visual modeling: a generic visual editor can be parameterized by process models, protocol models and artifact meta-models. Omega+ 'design environment' is a specific room where users can collaboratively create and customize models. Omega+ 'execution environment' gives access to all other rooms for executing the model-driven collaborative learning processes.

Omega+ client looks like a classical 'dual space' CSCL system [26], with a communication space including a protocol model-driven chat (similar to ωChat) and a task space. An information panel in the communication space displays textual or graphical information, in particular model-generated individual and group performance indicators. The task space may contain up to three tools as requested by the process model definition. Tools are either predefined editors (shared text editor, shared whiteboard) or shared graphical editors customized by artifact (meta-)models. Some components are adapted or extended for taking into account the specificities of dual space systems, like floor control mechanisms [27] and mechanisms for linking conversations and task objects [28]. For instance, in Fig. 5, the 'Circular Work' protocol controls the floor in both spaces (task + communication) in a round robin fashion until users explicitly pass the floor to the next user with a 'Pass' message. The task space includes a read-only text viewer (its colored background shows that interaction is not possible) displaying use cases created during a previous phase and two instances of the generic visual editor customized with UML collaboration and class meta-models. The student having the floor in both spaces (it is the case of Jack in Fig. 5) can create or modify UML diagrams corresponding to the use cases and comment these initiatives with 'Say' messages.

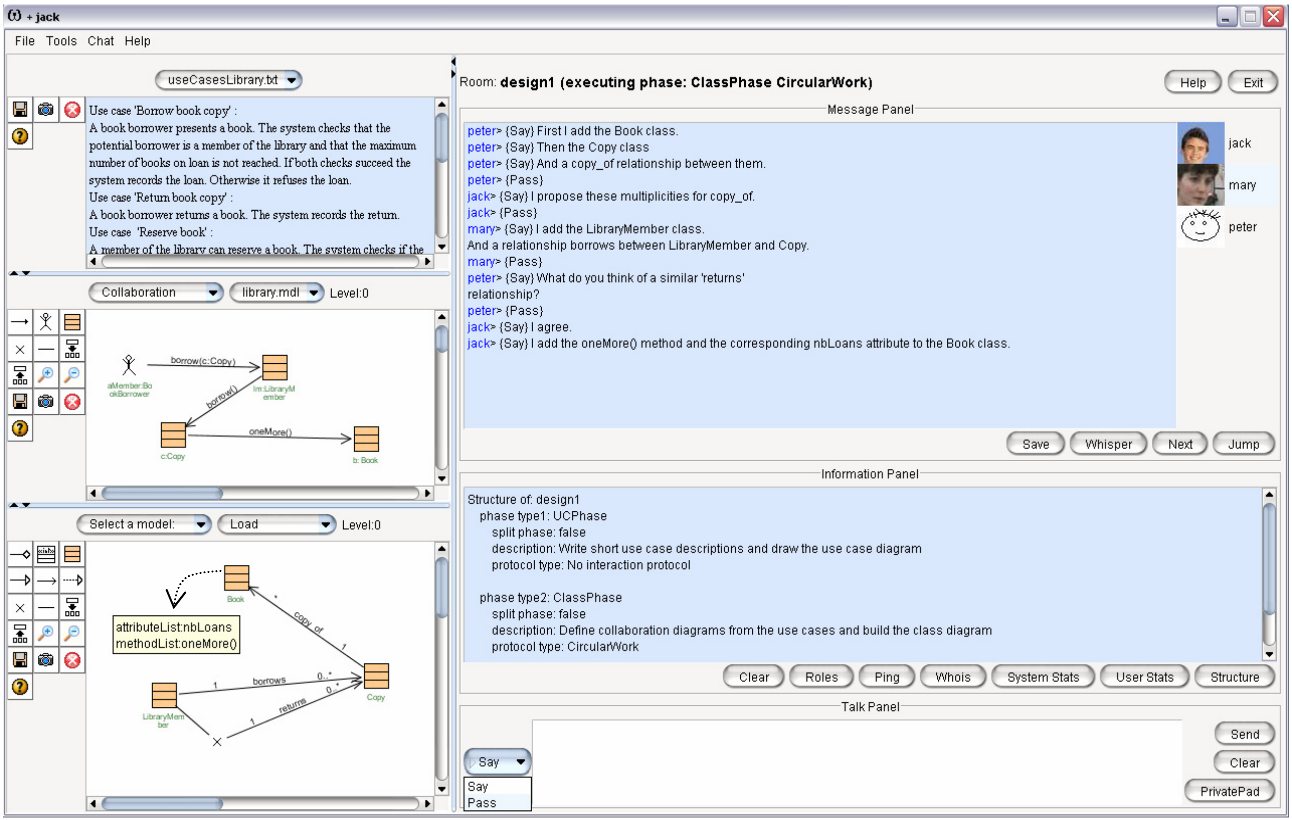


Figure 5. Jack's Omega+ client during a collaborative UML design process.

## *B. From ESCOLE to ESCOLE+*

ESCOLE+ is built on top of Libresource, an open source J2EE collaborative web platform (http://www.libresource.org). Libresource provides to its users a tree of projects, each project including a tree of documents and resources: wiki pages, forums, issue trackers, mailing lists, news, download areas, surveys, versioning tools, user groups (roles), timelines (event lists), project templates. Users can create, delete, move and modify resources and projects. Each project has its own security policy and access rights can be defined individually for each resource and role. New projects can be created resource by resource or by instantiating templates, i.e. predefined resource sub-trees. Projects, like all other resources, can be exported and imported as XML files. Libresource is built around a generic kernel that provides a naming service for managing all resources as a tree, an event service that can be used for awareness and services integration, a security service and a workflow service.

ESCOLE+ is organized in three spaces (projects):
(1) the 'Pedagogical Space', including a 'Community Space' for general information exchange and a 'Design Space' where Omega+ models are designed by teachers and CSCL specialists within dedicated sub projects,
(2) the 'Learning Space', where tutors and students execute model-driven collective learning processes within specialized sub projects,
(3) the 'Platform Space', for managing ESCOLE+ users, groups, documentations and so on.
Fig. 6 shows the root wiki page of the 'Design Space'. The menu on the left and the list at the bottom give access to all collective learning process definition spaces (the snapshot shows two definition spaces, called 'Class Diagram Design' and 'Summarization'). Each definition space is a design sub project, created from a standard template with a generic instantiation tool. Both are available in the left side menu. In each definition space, teachers and CSCL specialists can access Omega+ design environment. They can also use various communication tools for discussing all related pedagogical and technical issues (news, forum, issue tracker, wiki page, mailing list), and share technical documentations, experience reports, and Omega+ log files in the download area. For creating a specific execution space within the Learning Space, designers can work in the 'Libresource style' by creating manually a Libresource template in the design space and by using the dedicated instantiation tool. They can also work in the 'Omega+ style' by generating an XML project file from a Libresource process model visually created within Omega+ design environment.

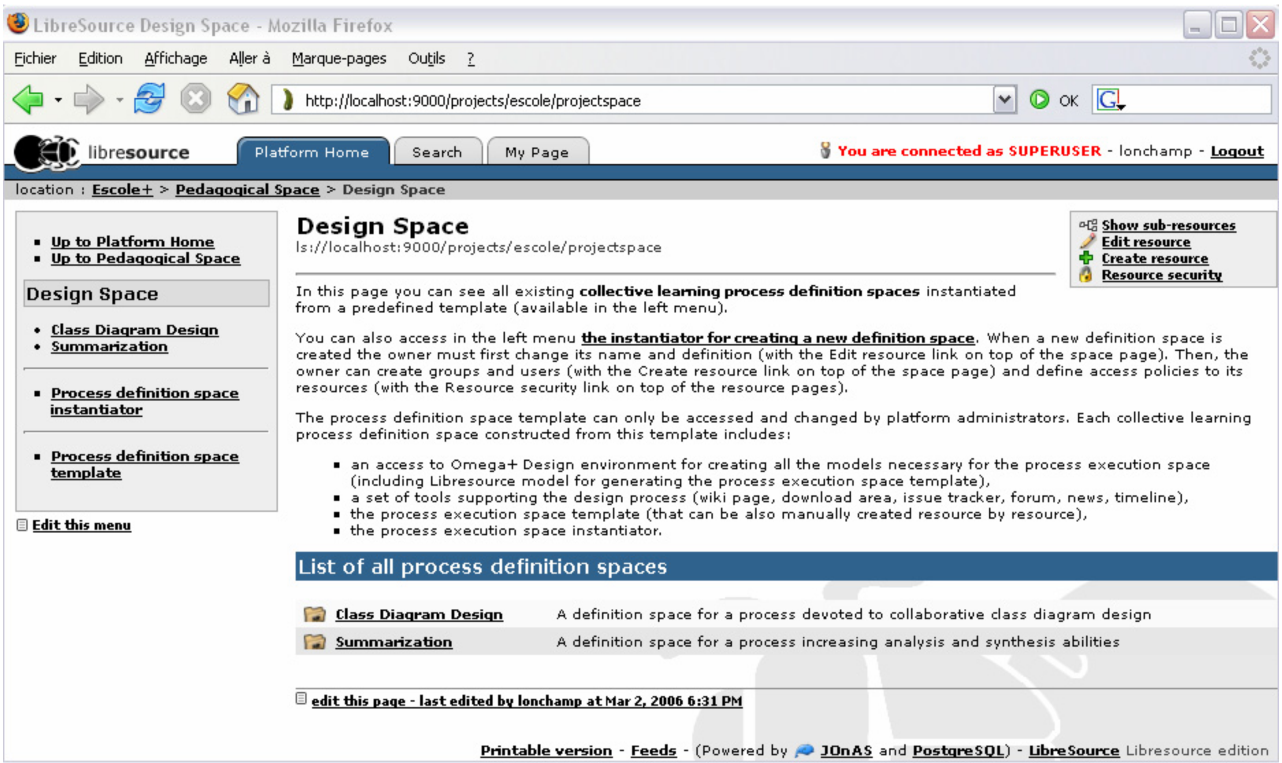


Figure 6. ESCOLE+ 'Design space'.

Fig. 7 shows a sub space of the 'Learning Space' (corresponding to the 'CSCL Session' in the 'Summary1' process) where students perform their activities. The menu on the left shows the process structure, its resources and user groups. Explanations are given in the central wiki page and specific news can be displayed at the bottom of the page. In some sub spaces students are cooperating asynchronously through asynchronous communication tools such as download areas, forums and mailing lists. In the 'CSCL Session' of Figure 7 students can call Omega+ execution environment for performing a model-driven synchronous session.

In summary, ESCOLE+ complements Omega+ in three domains. (1) ESCOLE+ provides web support for hybrid processes mixing synchronous and asynchronous interactions, like other recent CSCL systems such as KnowledgeForum [29] or Synergia [30]. (2) ESCOLE+ servers centralize detailed usage information, through Omega+ logs and ESCOLE+ event lists. For instance, it makes possible to measure the models which are chosen in the library, the percentage of teachers who try to customize them, and the features which are used by tutors and learners. (3) ESCOLE+ provides a comprehensive support at the meta-level (activities 1 to 4 defined in the introduction section). Additional facilities for replaying sessions are under development.

## V. Related Works

Online communities of practice in the service of teacher professional development, such as Tapped In 2 [31], share some similar objectives with the approach presented in this paper. However, our approach is focused on CSCL usage and provides a more effective support,

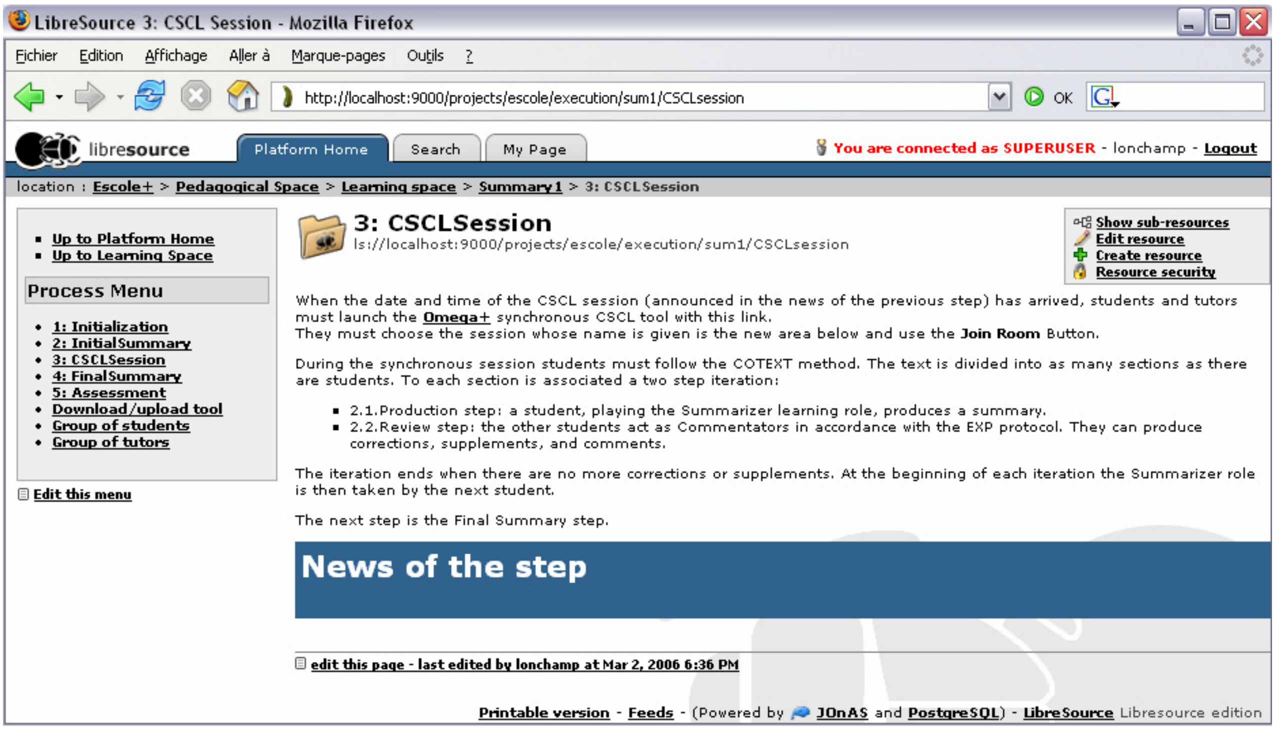


Figure 7. A learning (sub) space.

process-centered rather than information-centered. The concept of C3MS (Community, Content and Collaboration Management System) is also closely related to our proposal [32]. A C3MS extends a Content Management System (CMS), like Zope or PostNuke, with bricks for socio-constructivist scenarios, such as argumentation tools, project-based learning tools and scenario authoring tools. Unfortunately, a CMS tool is a weak basis for implementing the logical architecture described in section 2. For instance, the few existing CMS workflow services are specialized for web content production and cannot support arbitrarily defined processes. More fundamentally, some recent works propose to distinguish between ‘macro-scripts’ and ‘micro-scripts’. Macro-scripts define the composition of groups, the distribution of roles and resources, and the coordination of coarse-grained activities that make up a learning process [33]. Micro-script represents the fine-grained actions that each participant should accomplish within activities, for instance during argumentation processes [34]. This perfectly fits with our approach wich provides hybrid collective process support at the platform level and synchronous collaborative process support at the generic CSCL tool level.

## VI. CONCLUSION

This paper proposes a technological approach for making CSCL practice and dissemination easier through a workflow-enabled web platform hosting virtual communities of practitioners. Two successive versions of the platform have been developed with the aim of identifying and solving all problems related to customizability and flexibility on the one hand, and ‘teacher-friendliness’ on the other hand. Collaborative visual modeling, community support, workflow-enabled process deployment, model-generated meta-cognitive tools are example of elements that facilitate collaborative process design and execution by teachers.

The final objective of our research is to enlarge the community of CSCL practitioners far beyond the kernel of early adopters. But a long way is still to go for building more powerful and flexible CSCL tools. The example of open source software shows that a large exposure of tools to a community of practice is an efficient mean for meeting such qualitative goals. A platform for CSCL practice and dissemination could also help in that direction.

## REFERENCES

[1] E. Haatainen, and K. Korhonen, “Guidelines for teacher training and technical and pedagogical support: ITCOLE teacher training and consulting model”, ITCOLE Project Deliverable D8.1, Helsinki, Finland: University of Art and Design, Media Lab, 2002, http://www.eurocscl.org/site/itcole/D8_1_guidelines_for_teach.pdf.

[2] M. G. Moore, “Transactional distance theory”, In *Theoretical principles of distance education*, D. Keegan (Ed.). London: Routledge, 1993, pp. 22-38.

[3] M. J. Baker, M. Quignard, K. Lund, and A. Séjourné, “Computer-supported collaborative learning in the space of debate”, *Proceedings of the Int. Conf. on Computer-Supported Collaborative Learning*. Dordrecht, Netherlands Kluwer Academic Publishers, 2003, pp. 11-20.

[4] A. Soller, F. Linton, B. Goodman, and A. Lesgold, “Toward intelligent analysis and support of collaborative learning interaction”, *Proceedings of the Int. Conf. on Artificial Intelligence in Education*. Amsterdam, Netherlands: IOS Press, 1999, pp 75-82.

[5] M. Constantino-González, and D. Suthers, “Coaching Collaboration in a Computer-Mediated Learning

Environment", *Proceedings of the Int. Conf. on Computer-Supported Collaborative Learning*. Hillsdale, NJ: Lawrence Erlbaum, 2002, pp 583-586.
[6] D. Suthers, and D. Jones, "An architecture for intelligent collaborative educational systems", *Proceedings of Int. Conf. on artificial intelligence in education.* Amsterdam, Netherlands: IOS Press, 1997, pp 55-62..
[7] M. J. Baker, and K. Lund, "Flexibly structuring the interaction in a CSCL environment", *Proceedings of the Euro AIED Conference*. Lisbon, Portugal; Edições Colibri, 1996, pp. 401- 407.
[8] J. Jaspers, G. Erkens, and G. Kanselaar, "COSAR: Collaborative writing of argumentative texts", *Proceedings of the Int. Conf. on Advanced Learning Technologies*. Piscataway, NJ: IEEE Press, 2001, pp. 269-272.
[9] A. Dimitracopoulou, "Designing collaborative learning systems: Current trends and future research agenda", *Proceedings of the Int. Conf. on Computer Supported Collaborative Learning*. Mahwah, NJ: Lawrence Erlbaum, 2005, pp. 115-124.
[10] D. Suthers, "Technology affordances for intersubjective learning: A thematic agenda for CSCL", *Proceedings of the Int. Conf. on Computer Supported Collaborative Learning*. Mahwah, NJ: Lawrence Erlbaum, 2005, pp. 662-671.
[11] L. Lipponen, "Exploring foundations for computer-supported collaborative learning", *Proceedings of the Int. Conf. on Computer Supported Collaborative Learning*. Hillsdale, NJ: Lawrence Erlbaum, 2002, pp. 72-81.
[12] J. Lonchamp, "Supporting synchronous collaborative learning: a generic, multi-dimensional model", *Computer-Supported Collaborative Learning*, 2002, 1, pp. 247-276.
[13] N. Avouris, V. Komis, M. Margaritis, and C. Fidas, "Modeling space: A tool for synchronous collaborative problem solving", *Proceedings of Int. Conf. on Educational Multimedia & Telecommunications*. Charlottesville, VA: AACE Press, 2004, pp. 381-386.
[14] N. Pinkwart, "A plug-in architecture for graph based collaborative modeling systems", *Proceedings of the Int. Conf. on Artificial Intelligence in Education*. Amsterdam, Netherlands: IOS Press, 2003, pp. 535-536
[15] J. Börding, A. Voss, J. Walther, V. Wolff, A. Ocakli, R. de Groot et al., "DUNES-Dialogic and argumentative negotiation educational software", In *Proceedings DeLFI 2003, LNI 37*. Bonn, Germany: GI, 2003, pp. 290–298.
[16] R.K. Whitehead, and D. Stotts, D., "ProChat: Dynamic formal collaboration protocols in a chat tool for handled collaboration", Technical Report UNC TR00-016. Durham, University of North Carolina, 2000.
[17] H. R. Pfister, and M. Mühlpfordt, M., "Supporting discourse in a synchronous learning environment: The learning protocol approach", *Proceedings of the Int. Conf. on Computer- Supported Collaborative Learnin*. Hillsdale, NJ: Lawrence Erlbaum, 2002, pp. 581-589.
[18] A. Gogoulou, E. Gouli, M. Grigoriadou, and M. Samarakou, "ACT: A web-based adaptive communication tool", *Proceedings of the Int. Conf. on Computer Supported Collaborative Learning*. Mahwah, NJ: Lawrence Erlbaum, 2005, pp. 180-189.
[19] S. Farnham, H. R. Chesley, D. E. McGhee, R. Kawal, and J. Landau, "Structured online interactions: Improving the decision-making of small discussion groups", *Proceedings of Computer Supported Cooperative Work*. New York, NY: ACM Press, 2000, pp. 299-308.
[20] G. von Krogh, S. Spaeth, and K. Lakhani, "Community, joining, and specialization in open source software innovation: a case study", *Research Policy*, 32, 2003, pp. 1217–1241.
[21] S. George, "SPLACH: a Computer Environment Supporting Distance Project-Based Learning*", Proceedings of the World Conference on Educational Multimedia.* Charlottesville, VA: AACE Press, 2002, pp. 588-593.
[22] M.L. Bedbeder, and P. Tchounikine, "Symba: a Framework to Support Collective Activities in an Educational Context", *Proceedings of the Int. Conf. on Computers in Education.* Charlottesville, VA: AACE Press, 2003, pp. 188-196.
[23] T. Vantroys, and Y. Peter, "A WMF-based workflow for e-learning", *Proceedings of European Research Seminar Advances in Distributed Systems.* Bologna, Italy: University of Bologna, 2001, http://www.cs.unibo.it/ersads/papers/vantroys.pdf
[24] S. George, and P. Leroux, "An Approach to Automatic Analysis of Learners' Social Behavior During Computer-Mediated Synchronous Conversations", *Proceedings Intelligent Tutoring Systems, LNCS 2363*. Berlin, Germany: Springer-Verlag, 2002, pp. 630–640.
[25] J. Lonchamp, "A Structured Chat Framework for Distributed Educational Settings", *Proceedings of the Int. Conf. on Computer Supported Collaborative Learning*. Mahwah, NJ: Lawrence Erlbaum, 2005, pp. 403-407.
[26] P. Dillenbourg, P. and CSCL SIG of Kaleidoscope, "Dual Interaction Spaces", CSCL'05 workshop presentation, http://www.cscl2005.org/Workshops/workshop5.htm
[27] J. Lonchamp, "Floor Control in Complex CSCL Synchronous Environments", WebIST'07, in press.
[28] J. Lonchamp, "Linking Conversations and Task Objects in Complex Synchronous CSCL Environments", WebIST, in press.
[29] M. Scardamalia, "Knowledge Forum (Advances beyond CSILE)", *Journal of Distance Education*, 17 (Suppl. 3), 2003, pp. 23--28.
[30] G. Stahl, "Groupware goes to school: adapting BSCW to the classroom", *International Journal of Computer Applications in Technology*, 19(3-4), 2004, pp. 162-174.
[31] P. Schank, A. Harris, and M. Schlager, "Painting a Landscape onto TAPPED IN 2", CSCW Workshop on The Role of Place in Virtual Communities, 2002, http://tappedin.org/tappedin/web/papers/2002/TI2PlaceCSCW.pdf
[32] D. Schneider, P. Synteta, and C. Frété, "Community, Content and Collaboration Management Systems in Education: a new chance for socio-constructivist scenarios?", *Proceedings of the Pan-Hellenic Conf. on Information and Communication Technologies in Education*. Athens, Greece: Interactive, 2002, pp. 175-184.
[33] P. Dillenbourg, and P. Tchounikine, "Flexibility in macro-scripts for CSCL", *Journal of Computer Assisted Learning*, 23, 1, 2007, in press.
[34] A. Weinberger, F. Fischer, and K. Stegmann, "Computer-Supported Collaborative Learning in higher education: scripts for argumentative knowledge construction in distributed groups", *Proceedings of the Int. Conf. on Computer Supported Collaborative Learning.* Mahwah, NJ: Lawrence Erlbaum, 2005, pp. 717-726.

**Jacques Lonchamp** is currently full professor in Computer Science at Nancy University (France) and researcher at LORIA Laboratory (ECOO INRIA project). His main research interests include collaborative learning, collaborative work and software process modeling.