Towards a Methodology for Building Collaborative Learning Applications

Jacques Lonchamp, LORIA, Campus Scientifique,
BP239, 54506, Vandœuvre-lès-Nancy Cedex, France, jloncham@loria.fr

Abstract: Future collaborative learning technologies are characterized by the CSCL community as highly malleable and flexible. A promising approach for meeting these expectations is to use explicit models which parameterize a generic kernel for flexibly supporting different kinds of collaborative applications. This technical orientation raises complex issues related to the production of such operational models. A preliminary process for supporting non-specialist teachers in charge of designing collaborative learning application models is briefly outlined in this paper.

Introduction

A promising approach for building malleable and flexible collaborative learning environments is to use *explicit models* which *parameterize a generic kernel*. By providing ad hoc models, teachers can tailor the kernel to their specific needs (*definitional malleability*). Moreover, the behavior of the customized system depends on that, continuously queried, model and can dynamically evolve when the model is modified (*operational malleability*). This technological orientation, implemented in the Omega+ project (Lonchamp 2006), raises complex issues related to the *production of such operational models* by non-specialist teachers.

Modeling Collaborative Learning Activities - Omega+ Approach

An important requirement for effective collaborative learning is the combination of communication with shared work artifacts. Omega+ environment follow the *dual space* paradigm, by providing two distinct spaces of interaction. The task space is the place where students *do things*, by creating and manipulating task objects, i.e., elements of the shared artifacts. The communication space is the place where dialogue-based interaction, mostly textual, takes place. Students mainly *talk of what they do*. In a non-trivial CSCL application, the learning task is structured into a process including a sequence of collaborative phases. Within each phase participants can play different roles which constraint how they can act (in the task space) and how they can talk (in the communication space). Omega+ environment is parameterized by *four models*, corresponding to the *process dimension*, the *interaction dimension*, the *artifact dimension* and the *effect dimension* of collaborative learning activities. This approach makes possible to build the activity representation in different ways, adapted to the skills and needs of different categories of end-users: just reusing existing combinations of models, building new combinations of existing sub-models (i.e., following a very high level configuration process), defining or customizing sub-models through high-level visual languages, or through low-level specification languages (including programming languages). The rest of the paper focus on the third case and aims at supporting non-specialist teachers who design their own collaborative learning applications through high-level visual languages.

A Preliminary Model Design Process

A four stages process prescribes a sequence for formalizing progressively with high-level visual notations the main concepts whose semantics is important for defining and scaffolding a collaborative learning project. In the first stage, the *project tree* notation is used for highlighting the connectedness among individual components including (1) *objectives, (2) activities, (3) artifacts and (4) tools*. Figure 1 shows two possible decompositions of a collaborative UML class diagram design process and exemplify how the components are linked together. In the second stage, the project tree is transformed into a process model, i.e. a sequence of collaborative phases. Different kinds of transformation may occur. In many cases, *collaboration patterns* can drive this transformation independently of the task semantics. For instance, in the pyramid pattern (Hernández-Leo et al., 2005), a set of initial groups study a problem and propose a solution. Then, these initial groups join in larger groups in order to generate a common solution. At the end, all reunified participants propose a final and agreed solution. When transformed by the pyramid pattern, the '*one shot design*' project tree can give the process model depicted in Figure 2. Two dyads, working separately within *Design1* and *Design2* 'split phases' (i.e., parallel phases for sub groups), join in the *Final Design* phase. This process model is interpreted by Omega+ for piloting the learning process. In the third stage, the *interaction tree* notation documents the decomposition of the *community (1) into role types (2)* and *initiative types (3) associated to each role type*. Figure 3 shows two possible organizations for a design phase. In the first solution, called '*Circular design*',

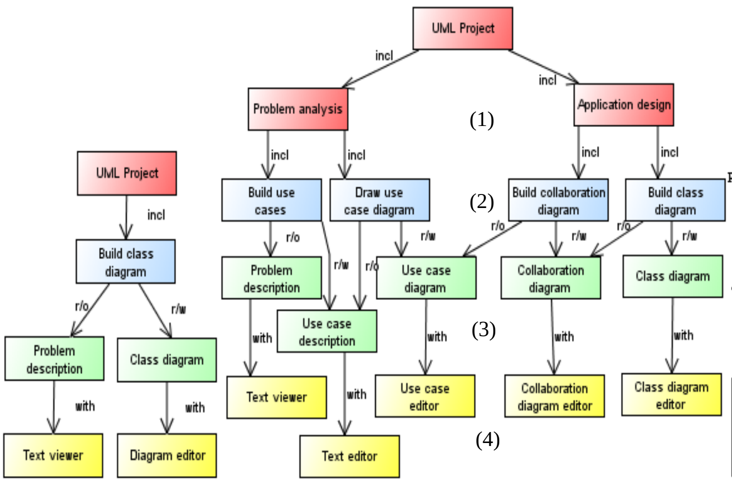

Figure 1. 'One shot design' & 'stepwise design' project trees.

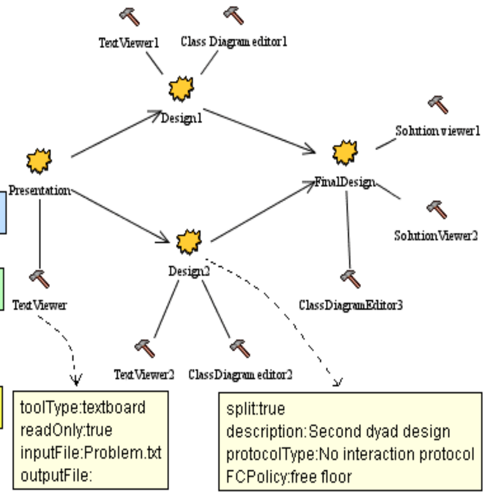

Figure 2. 'Pyramid' process model.

a single role type (*Designer*) can either *Do things* or *Pass the floor*. In the second solution, called '*Thinking aloud pair solving*' (Hernández-Leo et al., 2005), two students are assigned specific roles (*Solver* and *Listener*). The problem solver explains her/his solution to the problem (*Explain*). The listener follows the explanation and can suggest corrections (*Suggest* followed by *Accept* or *Reject* by the problem solver). In the final stage, interaction trees are transformed into fully-fledged Omega+ protocol models. First, teachers must choose if the protocol aims at controlling only the communication space or both the communication space and the task space. This is specified in the FCPolicy attribute of each phase. Then, teachers must specify how participant initiatives (utterances) are organized through the concept of adjacency pair. For example, the '*Circular design*' interaction tree is transformed into the interaction model of Figure 4, called '*Circular work*'. This protocol can control both spaces with the '*common protocol*' FCPolicy, ensuring a possible participation of each designer in turn. More details about floor control policies can be found in (Lonchamp, 2007a).

For supporting this model design process, a *collaborative design environment* is hosted into a web platform dedicated to *CSCL practice, evaluation, and dissemination* (Lonchamp, 2007b). This design environment provides some additional tools like a shared concept map diagram editor for discussing content-related issues.

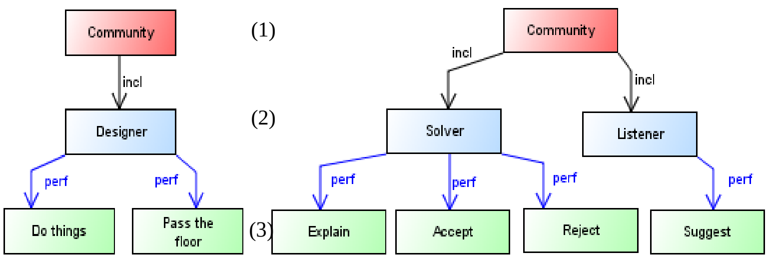

Figure 3. 'Circular design' & 'thinking aloud pair solving' interaction trees.

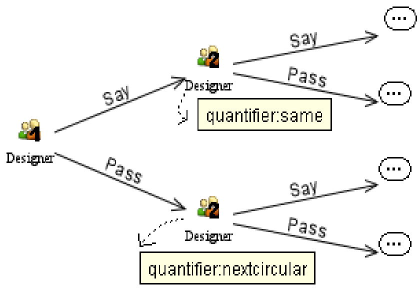

Figure 4. 'Circular work' protocol model.

References

Hernández-Leo, D., Asensio-Pérez, J.I., Dimitriadis, Y., Bote-Lorenzo, M.L., Jorrín-Abellán, I.M., & Villasclaras-Fernández, E.D. (2005). Reusing IMS-LD Formalized Best Practices in Collaborative Learning Structuring. *Advanced Technology for Learning*, *2* (3), 223-232.

Lonchamp, J. (2006). Supporting synchronous collaborative learning: a generic, multi-dimensional model. *Computer-Supported Collaborative Learning*, *1* (2), 247-276.

Lonchamp, J. (2007a). Floor Control in Complex CSCL Synchronous Environments. *Proceedings WebIST'07*. Barcelona, Spain, March 2007, INSTIC.

Lonchamp, J. (2007b). Towards a Web Platform for Collaborative Learning Practice, Evaluation and Dissemination. *Journal of Computers*, 2(5), 1-8.