CO-LAB, DESIGN CONSIDERATIONS FOR A COLLABORATIVE INQUIRY LEARNING ENVIRONMENT

Co-Lab is a learning environment for collaborative inquiry learning in which learners can experiment, make models, consult background information and discuss their findings. The complex learning processes in such an environment need to be supported by appropriate features in the software. In this paper we report on how, based on existing literature and specific user and usability studies, we developed a set of design guidelines that were implemented in the software.

Elwin R. Savelbergh, Universiteit van Amsterdam, The Netherlands
Ton de Jong, Twente University, The Netherlands
Wouter van Joollingen, Universiteit van Amsterdam, The Netherlands
Patrick Sins, Universiteit van Amsterdam, The Netherlands
Ard Lazander, Twente University, The Netherlands
Sarah Manlove, Twente University, The Netherlands

Introduction

Co-Lab is a learning environment for collaborative inquiry learning in which learners can experiment through simulations and remote laboratories, model, by using a dedicated modeling tool (http://www.co-lab.nl). Collaboration in Co-Lab refers to computer mediated collaboration in real time, i.e. a group of students are working together much like as if they were sitting together behind a single computer, but now with the students each behind their own computer screens, and all communications occurring over the network. This type of collaboration is intended to facilitate distance learning, collaborative homework and other flexible forms of delivery, as well as to provide a way of structuring and supporting the collaboration process. Co-Lab is intended primarily for use by upper level high school students in the domain of the sciences.

Learning processes in Co-Lab will be complex both in terms of the learning process itself and in terms of the collaborative process. In order to support these processes, first, we needed to develop a framework to describe these learning processes, and second, we needed an insight in problems that would occur and in the functions that must be fulfilled by the software. Therefore, we reviewed the already available software, we conducted user and usability studies, and we have held consultation meetings with teacher panels and domain experts. The user studies were conducted by having small groups of high school students (the target audience) work with (partial) prototypes of the system (Lazander et al. 2002). In this paper we will report on how these studies resulted in a set of design guidelines that were implemented in the software, and on the small scale usability studies that led to revisions of the design.

The collaborative inquiry learning processes in Co-Lab

In collaborative discovery learning, learners will perform various different learning processes that together constitute the whole process of collaborative inquiry. From the content perspective, the main process can be said to be inquiry, which involves a cycle of
orientation, hypothesis building, experimenting, and drawing conclusions. Computer
modelling is a way to implement these phases. In order to coordinate the inquiry process,
a good deal of (collaborative) planning and monitoring will be necessary. Figure 1
provides an overview of the relevant processes (De Jong, Van Joosen, Savelberg,
Lazonder, Wilhelm, Ootes, 2002).

Finally, in defining the Co-Lab learning process, the word ‘collaborative’ needs further
attention. What we are aiming at in Co-Lab is an increased conceptual understanding in
all group members, rather than a perfect computer model as a goal in itself. Therefore, it
is not sufficient that students cooperate to achieve their goals. For instance if subtasks
would be distributed, this could be an effective way of working, but not an effective way
to ensure understanding for all. Therefore, although at some points in the working
process students may work independently, at crucial moments in the process they need to
convene in order to ensure a common focus and common understanding.

Identification of issues and design solutions in Co-Lab

A first challenge in such a complex learning environment with multiple modules,
multiple tools and multiple students trying keep track of each other, is to support
navigation. As a first attempt to translate insights from theory, we implemented an
animated mock up of the system (Figure 2). In a series of mock-up designs, we
developed a ‘city metaphor’ to structure a student project in modules (buildings),
submodules (floors), and activities (rooms). In this way, buildings and floors organise
the domain, rooms organise the learning activities. The basic unit of group activity is the
floor. At each floor we have a laboratory room (for experimenting), a theory room (for
creating models) and a meeting room (for overall planning and monitoring of the learning
process). Students in a group can move individually between rooms, which allows both
collaboration on a single activity (if several students are in the same room) and
cooperation between students working on different subtasks (in different rooms). In addition to providing different modes of collaboration, this way of organizing Co-Lab in rooms floor and larger units also provides a means to control the balance between guidance and freedom, as students can be given complete freedom to organize their working processes at a small scale (e.g., move between rooms on a floor), while at a larger scale (e.g., move to a next floor) decisions can be subject to rules set by the teacher or by the system.

A further problem is that if all students are collaborating in a single room it can be counterproductive if all group members perform their actions simultaneously. Although for many actions it is fine if students act simultaneously (e.g. chat), for other actions it may be desirable that the others do not interfere, or even pay attention, when one student is performing an action (e.g. editing a formula). In a face-to-face situation, with two students working behind one computer, this problem will not occur because only one student will hold the mouse (the risk being that the student will keep holding the mouse and dominate the process). In order to structure roles and responsibilities, a leadership model was introduced in Co-Lab where in each room one student at a time would be in control, and control could be passed from one student to another.

At a more detailed level Co-Lab accommodates a number of cognitive tools to support specific learning processes, such as the modeling process, the planning and monitoring process and the collaboration and communication processes.

From the first user studies, it became clear that in the initial design students in a group tended to lose track of each other’s presence, status, and location. Therefore, in the revised implementation (Figure 3), in order to enhance mutual awareness, the original simple traffic light, was replaced by a status panel, where students could monitor the whereabouts of the other group members. Moreover, the original chat function, which was only between students in the same room, was enhanced with a ‘shout’ function, which would also reach the students in the other rooms.

If students are to develop and discuss their models collaboratively, these models must be represented in such a way that discussion about these models is supported. More in particular, the model representation must allow both global discussions about the conceptual structure and more detailed discussions about computational details of the

![Figure 2: Animated mock up](image1.png) ![Figure 3: Revised implementation](image2.png)
model. While a graphical qualitative representation proves to be better suited to support the first aim, a quantitative formula based representation is needed to fulfill the second aim (Löhner, Van Joologen, and Savelberg, 2003). We concluded that a graphical ‘system dynamics’ model representation, like the one used in STELLA might allow both, provided that it would be enhanced with a semi quantitative mode, somewhat similar to Modellt. The model editor was implemented according to these ideas, and is currently being tested (Figure 4).

If students are to set up a scientific argument, which deals with hypotheses, data, evidence, and counter evidence they run into many difficulties (Chinn and Brewer, 2001). Therefore, it was expected that students would need support in setting up such argument, and we sought ways of structuring their talk. We found an example in the Belvedere argumentation software, where statements are organized in a tree diagram, and were each statement carries a label (e.g., question, argument, or hypothesis). It was expected that if we would provide similar labels in the chat, students might use these to indicate with each chat message what the functions of their statements was. In studies with a partial prototype of the system, it appeared that this feature did not help because it was not used by the students (Lazonder, Wilhelm, and Ootes, 2003).

![Figure 4: model editor, semi-quantitative mode](image1)

![Figure 5: the process coordinator goal tree with issues and hints.](image2)

Also with respect to the larger process of planning and managing their inquiry processes, it is known from previous research that students tend to skip essential phases and fail to monitor their working process, leading to ineffective learning processes (De Jong and Van Joologen, 1998). To help students structure their inquiry process, we had introduced a process coordinator tool, which helps students identify phases in the inquiry process and keep track of their progress. Based on both the existing literature, and our own user studies, it became clear that while at the one hand we would like to emphasize the generic structure of the inquiry learning cycle, at the other hand purely generic structures provide little value for the learners, and there is a clear need for specific support. This was resolved by having a generic structure of goals and subgoals, which could be augmented by domain specific issues and hints (Figure 5). The process coordinator as used in Co-Lab shares characteristics with the cooperation script as introduced by Kollar et al. (this volume). In addition to planning, students also needed support for monitoring during the
process, and for reflection after the process. This consideration led to a design where the process coordinator would not only provide an outline of goals and hints, but also would serve as a log where students would put in their comments and enter their products once they were satisfied. Furthermore, in order to have the students look back on what they did, we also included a report writing function in the process coordinator, where students could use their notes and products from the log to compose a report. As a side effect, these features also provide a teacher with an opportunity to track the students’ working process.

Conclusion

Based on the above considerations the Co-Lab software has been implemented. In addition to the above, there are of course many features of the system that have been designed based on the designers’ implicit understanding of learning processes, often based on anecdotal evidence. In addition, where most user studies addressed specific tools, the usability of the system as a whole cannot be addressed before the system has been implemented. Moreover, whether an environment such as Co-Lab provides an effective learning environment does not only depend on the user interface design but also on the availability of appropriate contents. We are now in the process of implementing the contents, and fine tuning the system so that longer learning sequences can be observed.

Bibliography


