Is there a way to e-Bologna? Cross-National Collaborative Activities in University Courses

Andreas Harrer[+], Giorgos Kahrimanis[*], Sam Zeini[+], Lars Bollen[+], Nikolaos Avouris[*]

University of Duisburg-Essen[+], University of Patras[*]

Abstract. This article describes a study of distance collaborative activities that have been conducted in a cross-national setting between a Greek and a German university. We discuss issues related to organization, technology, and curricula considerations. In addition, we analyze the modes of cooperation that have been chosen in the students' work on creative problem solving tasks and conclude that for complex learning scenarios succesful collaboration and peer tutoring in advanced learning support environments is possible, but requires careful preparation and planning. Further we draw conclusions on possible wider implications for such approaches in the emerging common European Area of Higher Education in the frame of the Bologna process.

Introduction

Today's work reality involves in a growing degree multi-national activities engaging individuals in remote interaction and collaboration. As a consequence, learning activities, especially at the Higher Education level need to be expanded accordingly in order to prepare students for distant work and cross-cultural communication. This is especially important for the area of computer science, where global projects and multi-national enterprises are a common phenomenon. Thus it is very valuable, both for the students and also for the academic staff, to find opportunities to allow students to experience the challenges, problems, and potential of remotely conducted projects in a multi-national and/or multi-cultural setting. It should be stressed that identification of the conditions for these opportunities and creation of new opportunities for the cooperation of educational establishments across national borders in Europe is one of the main aspects of the Bologna process. The Bologna declaration aims at the creation of a European Area of Higher Education through collaboration among high quality higher education institutions to enforce student mobility, competency development and educational co-operation (Bologna Secretariat, 2006). In this context, studies which focus on the potential of technology enhanced learning for the integration of the European university systems and the strengthening of reliable computer-supported cooperation between institutions in different European countries are of particular importance. In this article we present the design and implementation of such study that engaged remotely students of the University of Patras in Greece and the University of Duisburg-Essen in Germany.

Researchers of the two Universities have been engaged during the last years in study and development of collaboration support technologies. These have been made available to the Academic staff and the students who participated in the study.

In the rest of the paper we present first the organizational and educational framework of the study, followed by a brief description of the technological infrastructure used. Subsequently we discuss the main findings of the study as they emerge from analysis of logs of interaction of cross-national group activities and attempt to draw wider conclusions.

Organizational and Curricular Framework

The scenario that was implemented during the study was closely connected to the teaching practice of both parts, i.e. students in the context of real lectures are given short exercises as homework. Usually such exercises are performed individually. In our case they were requested to work in dyads with a partner across the border, using provided collaboration support tools. This scenario left a lot of freedom to the students to approach their task, in terms of when and how to work together or how to divide the work. The use of a more contained lab situation might have been preferable for controlled analysis of some collaboration aspects, but would have produced an artificial environment that would not have been connected well to the students' real world experiences.

In order to proceed with a detailed plan of the activity, we first had to explore the viability of a joint study with respect to curricular opportunities for cooperation and the organizational affordances of it.

To avoid severe grounding problems (Clark and Schaefer, 1989) the students had to have prior knowledge that can be connected with the partner's knowledge, thus it has to overlap or at least to be in each student's zone of proximal development (Vygotsky 1978, Murray & Arroyo, 2002). For this end we had to identify courses that were taught at the same time at both partner's sites, in which students could interact and jointly do an assignment. Because of similar profiles of both groups in research and teaching, we managed to identify two opportunities to conduct studies during the last year: The first involved courses in the area of Human-Computer-Interaction for advanced students (5^{th} year in Greece, 3^{rd} year in Germany with quite comparable background knowledge); this opportunity was used for a pre-study to test the setup of technical, organizational, and analytical aspects for a full-fledged study. This first study took place in the summer of 2005 with 9 mixed dyads of Greek and German students. While the joint work was mainly performed cooperatively in asynchronous mode, we obtained valuable insights in the usage of the provided tools and the organizational and technical setup for studies on a larger scale.

The second case involved introductory courses in computer science (the 1st year course for Greek students and an algorithm course for 2^{nd} year German students). In this case we could involve a more substantial number of students for the joint study. Here the background knowledge was well suited for information exchange and grounding, since both populations were more familiar with different aspects of our planned task: the Greek students were quite knowledgeable with diagrammatic representations of algorithms, while the German students were familiar with recursion, iteration, and the transformation between these two concepts. All the courses were conducted in the mixed lecture and lab/exercise format, usually also with online material and communication, which prepares the students nicely for the remote collaboration with the tools that were proposed to be used.

In addition to these curricular considerations such a joint study requires careful preparation of the organizational aspects, too. As the students initially do not know their remote partners, we thought that uncertainty about their role and limited awareness and familiarity with such a task might hinder the whole process. To compensate for this we decided to establish a discussion forum, with a general area for all the participants of the respective study and a specific area for each collaboration dyad. This forum was used for official announcements, the distribution of the task assignment, and questions and answers related to the planned process.

It was thought that there was a need to provide the mixed pairs with ample time to get acquainted with each other. So the time to deliver a solution was at least a full week in both studies. This made the task more realistic with respect to our usual style of assignments that also have one week periods for delivery.

Since the students tend to choose their own pace of work, the replies to their questions need to be provided in a timely manner, an issue that was proven particularly critical in this setting, for avoiding frustration and defection within the groups. In case that problems, especially technical ones, arise, a fast reaction by the support staff is required to enable the students to work on the task and not to fight technical problems. In our experience this is especially important in the phases close to the delivery deadline, because a lot of the collaboration groups tend to start late, even with getting acquainted with the tools. Another issue with the organization of such a study is that the technical environment used, has to be robust, especially with respect to persistent and regular data storage, since a data loss will demotivate students and endanger the whole learning experience considerably.

In the next section we proceed with a brief description of the environments used in the study and their technical characteristics.

Technology to Support Co-constructive Distance Scenarios

During our study, the students used a variety of collaborative modeling and communication tools. These tools were either external third party applications (e.g. *ICQ, MSN, Skype*), or available collaboration infrastructure (e.g. *phpBB* based forum) or specific tools that have been developed by the respective Universities to support synchronous collaborative activities (*Synergo* of the University of Patras and *FreeStyler* of the University of Duisburg-Essen). In the following, these tools are described in more detail.

General Communication Infrastructure: The Patras-Duisburg Web Portal

A forum was established as a source for general information about the studies, as a first meeting point for the dyads and as starting point for the use of other tools, especially for downloading and getting information about Synergo and Freestyler. The open source phpBB[1] bulletin board was used in this case. This forum consisted of a general discussion space for general questions and comments about the tools and the task, and public and private discussion spaces for each dyad. The forum was also meant as an option to contact the staff as fast as possible in case of organizational or technical problems, since a fast reaction time is crucial in these settings, as discussed above.

Specific Collaboration Tools

Two tools have been used in the studies to support the students' modeling tasks: *Synergo* and *FreeStyler*. In figure 1 typical screenshots of the two environments are shown. Both tools provide a shared workspace to allow for synchronous collaborative modeling activity, using various visual representations. In addition, both tools provide a chat facility to support direct communication between the partners. Each dyad of students in our study was provided with one of the two tools. One of the objectives of the study was to examine if the use of different tools may impact the collaborative activity and the quality of the produced work.

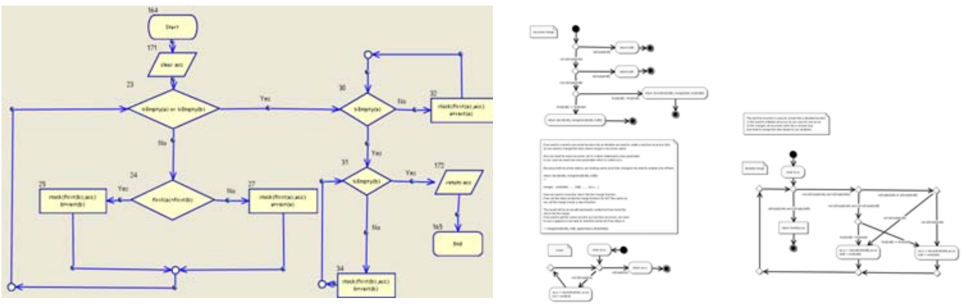

Fig. 1. Synergo and FreeStyler solutions of the given problem

FreeStyler (Gaßner, 2003) and Synergo (Avouris et al., 2004) bear many similarities. They both enable distance collaborative modeling and discussion scenarios. They are based on a shared workspace architecture, i.e. in a collaborative session, every participant can see all modifications to the workspace done by any participant directly, following the WYSISIS ("what you see is what I see") principle. Both tools are built to support collaborative activities by small groups of students and have been used in Secondary and Higher education settings for teaching various subjects (Lingnau et al., 2003).

However since they have been produced independently, they have distinct characteristics in terms of architecture, user interface design and performance:
FreeStyler allows modeling withuse of various visual modeling languages that can be added to the application withthrough a pluginplug-in mechanism. At the moment, there are around 30 pluginsplug-ins available, e.g. for creating Petri Nets, Concept

[1] See http://www.phpBB.com for further information.

Maps, models in System Dynamics models, Activity Diagrams, State Charts, stochastic experiments etc. In particular, elements of different visual languages can be used in the same workspace at the same time without loosing possibly operational semantics (e.g. the capability to actually run a System Dynamics simulation). In relation to the task mentioned hereof our study, the suitable plug-in that Freestyler provides is the one for *Activity Diagrams*, the UML diagrams for representing the flow of activities involved in an algorithm,. Activity Diagrams are very similar to *Flow Charts,* a more traditional visual representation of an algorithm.

On the other hand, Synergo provides a smaller number than Freestyler number of fixed libraries for diagram building, which are integrated in the environment by default. Related to the task, Synergo contains a *Flow- Chart* diagram notation library.

What also discriminates the two environments, is that the synchronous dialogue tool that they both provide, is integrated within the shared workspace in the case of Synergo, whereas it appears as an object in the shared workspace in Freestyler.

The shared workspace architecture of Freeestyler is based on the so-called MatchMaker (Jansen, 2003), a Java RMI-based synchronization middleware with a central server and a replicated data model. Synergo follows a similar approach; in this case the synchronization component is called Relay Server.

Finally, Synergo provides built-in tools for analyzing log files produced during Synergo activities. It provides automated measurements of logging events, coding of dialogue data, play-back of the activities as they were developed, and other features that are useful for analyzing data. Most of the analysis of the Synergo sessions was conducted with the aid of these tools.

Other tools used

Besides the tools mentioned above, the students were free to use any other tool that they might consider useful. They mainly used tools to support communication like ICQ[2], MSN Messenger[3] (chat tools) or Skype[4] (free audio conferencing tool). It was not possible to control or monitor the usage of these tools, but we asked the students to hand in copies of their communication logs (as far as possibly) together with their solutions, since they had to provide a proof of their collaborative effort.

General technical requirements

To conduct this study, several technical requirements have been crucial and needed to be considered beforehand. Some of them are mentioned briefly in this section.

In such diverse multi-national setting, one may face several unexpected problems related for instance to availability of the software in the needed languages, capability of the software to run on the operating systems that are used by the target groups, restrictions on use of the specific software due to security reasons (e.g., some establishments may not allow third party chat tools or have very strict regulations on computer usage regarding security issues.) or other technical difficulties that may arise especially in the case of research prototypes, related to special needs for software installation (e.g. the help of an administrator).

[2] See http:www.icq.com for further information.
[3] See http://messenger.msn.com for further information.
[4] See http://www.skype.com for further information.

Distant learning setups naturally make use of computer networks. But the setup can be rendered useless if it does not take into account issues like firewalls, NAT (Network Address Translation) schemes or bandwidth limitations that make it impossible to setup a adequate network connection between the participants.

In general, software failures or the loss of data can reduce the students' motivation drastically. Although one cannot expect research prototypes, like the Synergo and FreeStyler tools used in our studies, to run with no problems in such complex and diverse learning settings, one needs to decrease the impact of technical problems by e.g. providing robust servers, which are able to restart themselves in the case of a crash and by creating persistent data storage. Both approaches have been realized for this study and they turned out to work well for this kind of scenarios: Although a server crash did happened, the system remained operational without loss of data in either of the two parties and tools used.

In the next section we discuss the context of the study and the findings produced by analysis of the activity logs and the students work.

Case study: Joint construction of an algorithm

As a result of the pre-study (see above) we discovered that successful collaboration in advanced learning support environments is possible if properly supported by the setup. For the follow-up study we decided to evolve this hypothesis towards the problem of peer tutoring: with respect to the heterogeneity of the cultural and educational background of the peer students we assume that collaborative modeling and discussion tools can help to overcome the challenges that cross-national learning scenarios impose. Furthermore, by choosing an exploratory design, we aim at finding out and describing *how* such learning processes manifest themselves.

The reported study was conducted during the winter semester of 2005. 16 mixed dyads were engaged in the activity, consisting for the German University side of computer science students of third semester in the course "Algorithms and Data Structures" and for the Greek University of computer science students of first semester in the course "Introduction to Computers". Since the lectures were differently conceptualized and had overlapping but not exactly the same contents, we expected a challenge for the peers to find common ground with respect to the task. Thus we expected an initial discussion and peer tutoring phase in various aspects, such as diagrammatic notation and algorithmic knowledge. We also had a similar expectation in the knowledge about the tools and their usage, because the students in Greece were familiar with Synergo, while the German students knew FreeStyler from the lecture.

Half of the dyads were assigned to use FreeStyler while the others to use Synergo as the modeling tool. While, as discussed, both tools provide the possibility for integrating chat in the collaborative workspaces, yet the students were also free to use other chat or communication tools while they were engaged in collaborative activity. Some of them also used e-mail, the forum, and the forum's private messages functionality.

From the 16 teams that were originally formed to participate in the task, 10 managed to complete successfully the assignment. From them 7 were requested to use FreeStyler and 3 Synergo. However, 3 teams that were supposed to use Freestyler, finally decided to use Synergo instead. One reported that they used Synergo because the students were not familiar with Activity Diagrams and preferred to develop their solution in the form of Flow Charts, while the other two faced technical problems with the collaborative use of FreeStyler, and they decided to use Synergo instead. Freestyler had been tested successfully in previous occasions (Lingnau et al., 2003), so the technical problems are attributed to the special technical context of this activity, like the special characteristics of the network infrastructure in Greece. So finally the distribution of the collaboration tools was: 6 dyads used Synergo and 4 used Freestyler. An analysis of the log data of synchronous collaborative activity using these tools is included in the following section.

In addition to the two collaboration tools, a forum was provided so that the students would be able to communicate in pairs or publicly with all other participants. Just two out of ten pairs used this forum. One group used the forum in order to negotiate an initial solution to the problem that was provided in pseudo code. The same group also used the forum in order to plan and arrange their sessions. The second group used the forum just to coordinate the initial phase of interaction.

Alternatively, e-mail communication was used by half of the pairs in order to communicate and interchange initial solutions to the problem in the form of pseudo code and arrange their sessions. Surprisingly, half of the students reported that they communicated through other instant messaging applications as well, namely MSN Messenger, ICQ and Skype chatting.

Methodological approach

The log files that were collected were analyzed in order to examine patterns of behavior of the different dyads. One particular objective of this analysis was related to the effect of the collaboration tools used. These tools have not been compared formally, so we preferred a mainly qualitative approach enriched by some quantitative measurements. This enables us to interpret the data in the context of the respective tool used and also in a deeper level of detail. A qualitative data analysis software was used in this case. For the quantitative measures we used analysis tools that have been built by the respective research groups. In the case of Synergo the built-in analysis tools (Avouris, et. al., 2004) produced the indicators for the quantitative measures. The FreeStyler files were analyzed by the Pattern Discovery tool described in (Harrer et. al. 2005).

As a part of the analysis of the activity, we conducted a *content analysis* of the dialogues conveyed through different communication tools used. We assigned exchanged messages into categories provided by a content analysis schema that was developed. The purpose of the analysis was to provide an insight on certain aspects of the activity in order to interpret patterns of observed behavior. Content analysis was also useful as a heuristic tool that helped us to focus in parts of the activity that were of most interest. The schema developed is generally constituted of two main sub-schemata: one related to non task interaction and the other to task related interaction.

The first part of the schema, illustrated in table 1, classifies verbal interactions that do not relate to the task per se. Coordination and planning of the activity, comments on technical aspects or social conversation were included in the schema in order to find what kind of information is conveyed through the tool.

CATEGORY		DESCRIPTION OF CASES	EXAMPLE
Coordination		Arranging sessions, planning work, distributing tasks	"Should we meet tomorrow at 12:00?"
Technical	Functionality	Reporting or discussing bugs and inconsistencies noticed in the functionality of the collaborative tools.	"The session was disconnected"
	Usability	Reporting usability problems of the collaborative tools or difficulties of the students to get accustomed with the user interface functionalities	"How can I put text near the connector?"
Social		Social talk, encouraging collaboration	"Good Work ;-)"

Table 1: Coding scheme for non task-related categories

The second part of the schema concerns task-related messages. It refers to instances of *cognitive presence* (Garrison et. al. 2000). The rationale behind that model was inspired by the Interaction Analysis Model proposed by Gunawardena et. al. (1997).

However, the use of the model differs from Gunawardena's work in some important aspects. Firstly, our study concerns synchronous communication via chatting tools and activity in shared workspace instead of asynchronous communication through forums. Furthermore, the object of knowledge is not the negotiation of meaning of some abstract concepts, but the comprehension of algorithm building using visual notation and the development of algorithm development skills instead. This part of the model is illustrated in detail in table 2

CATEGORY	DESCRIPTION OF CASES
Phase 1: Communication of Information	Statements of observation, communication of knowledge about a certain issue, description or realization of a problem, asking questions for clarification, providing examples
Phase 2: Discovering Inconsistencies	Statements of disagreement on a proposed issue or solution to a problem, response to disagreement statements indicating a reform of the initial standpoint or provided solution
Phase 3: Co-construction of knowledge	Reaching consensus, proposing a (partial) solution that integrates new data inspired by contradictions of initial standpoints
Phase 4: Testing jointly constructed solutions / conceptions	Testing the result of the previous phase against certain examples, similar problems, trying to optimize the solution
Phase 5: Further application of new knowledge / meta-cognitive processing	Reflection on the knowledge gained, association to other knowledge, application of new knowledge in similar domains

Table 2: Coding scheme for task-related verbal interaction

Concerning the analysis process, what discriminates the model used from Gunawardena's proposal is that not all indications of certain phases are contained in dialogues, as indications of the presence of them can also be found in the shared workspace.

In the model used, it is assumed that knowledge construction evolves from lower to higher cognitive functions as proposed by Vygotski (1978). Similar distinction can be made in group skills during a collaborative activity (Smith, 1994). Hence, actions related to higher phases of the model are expected to follow lower phases. However this sequence is not strict, because not all phases have to be manifested in chat dialogues as already discussed. Furthermore, in the tasks analyzed, there were several learning objectives, namely comprehension of flow-chart diagrams, understanding the concept of recursion and the systematic development of algorithm solutions.

During the coding process new categories were detected in the data and introduced to the coding scheme by open coding. Frequently detected codes were discussed between the research partners aiming to find out which new categories would be common to both research approaches in the sense of a generalization process. The category *coordination* for example originally belonged to the global category "non task related". But during the coding we detected that numerous parts of the analyzed discussions were about planning task related issues. So we decided to distinguish between these two different types of coordination activity.

In the case of FreeStyler group the tool atlas.ti[5] was used for coding the conversation data. The conversation data came in this case from messenger chatlogs, forum postings and data extracted from FreeStyler logfiles, since FreeStyler also provides its own chat functionality. A pattern discovery tool was also used during the analysis process. In this case we just counted different types of workspace activities to compare them with the findings in the communication analysis.

The coding of non-task related categories was straightforward. Concerning the task-related categories, emphasis was put on the presence of each category throughout the activity. For example, in a certain phase, several messages indicated appearance of stage 2 of the model. Concerning the message as a unit of analysis, it was not considered important the number of messages that were coded according to it, instead, what seemed to matter was just the fact that e.g. a "discovering inconsistencies" episode occurred in a certain phase of the activity. So, annotated messages did not serve for measuring frequencies of occurrences, rather they were useful for locating most important episodes that would be further analyzed.

Concerning the Synergo group logs, the dialogues were coded using Synergo's analysis tools. Synergo provides a tool for developing a coding schema and annotating events according to it. Summative measures and measures in relation to time intervals are automatically calculated and related graphs are automatically illustrated.

Results of the study

Phases of problem solving activities: The problem solving process was similar in many groups and can be generalized to a phase of grounding, explanation, knowledge construction, coordination, evaluation and modeling. In the following a discussion and examples of each phase are included:

[5] http://www.atlasti.com/

Phase of Grounding. In this phase the peers checked out each other's knowledge by asking each other and trying to develop a common language, e.g. negotiations about definitions. These phases were indicated by the category "communication of information". An interesting example which shows that students also use advanced techniques for grounding is the one in which a Greek student developed a flow chart to explain his understanding of lists:

GR[6]*: "the thing is, i'm not really into lists (we only heard about them a month ago) so i'm not sure. if you could please wait, I can scan the flow chart I made for you to see."*

Phase of Explanation: These phases were observed very often while peer tutoring. One of the students who knows a concept tries to teach it to the other student. According to Gunawardena's model this is part of "Communication of Information". A typical example for such a case is where a student teaches recursion to the other student:

"DE: as I explained in the other session (Team-2 new2) you see that we need a third parameter
GR :i see"
DE: maybe I should begin and you see what i want to explain to you"

Construction of Knowledge: Theses phases describe the situations in which both students develop common knowledge by the collaborative task. An example for collaborative knowledge construction while modeling the different branches of the function to be developed is:

"GR: now we shall examine what happens if list a is empty
DE: its better than we havent so much recursive calls
GR: no..let's examine what happens if both lists are not empty first
DE: and if u ask not(isEmpty(listA)) then you can do samehting but you never know if listB is not empty.
[Communication break by workspace action] i think its right how you did it know
GR: you think we should change this?
DE: if both are not empty you have to look which value is bigger and stores this in the accu. [Workspace activity] right"

Coordination: As mentioned above we detected some cases of task related planning which comply to the Communication of Information according to the category scheme (Table 2). An example for planning can be shown for the case in which the students decided to take turns while modeling the task:

"GR: nice...your turn to make the if list b empty thing
DE :) [Workspace activity]
GR: nice"

Evaluation: This code is described by the fourth category in the coding scheme (Table 2). These cases were observed frequently in relation to using of pseudo code, for example:

[6] In dialogue extracts here forth, Greek student shown as GR, German student as DE

:DE: "At the point "stock (first (y)) y<--rest (y)" The operation stock; Doesn't we need two parameters. The first one for the element that we have to add and the next one the list it should be added to?

*GR: I think you're right*JOKINGLY*. that needs more thought now..."*

Use of auxiliary means during problem solving: Most partners attempted to solve the problem by expressing the solution in an auxiliary form, i.e. in pseudo code or in a programming language (Java) first on their own and then transform it to a flow chart or activity diagram collaborating with their peers. In most cases, both partners had developed or found a solution in pseudo code and presented it to the other at the beginning of the collaboration through Synergo / Freestyler and they interchanged the solutions via mails or forum postings.

In some cases the partners developed or improved their solution on paper while they had real time communication through the tools. Messages like: "I think I have still another idea. I write it down [on paper] at the moment" during a collaborative session indicated such cases. There were also cases that a partner presented the pseudo code solution in the form of a sticky note in the shared workspace, and testing and modifications were done collaboratively.

The number of collaborative sessions that were needed in order to accomplish the task, and the duration of each session varied among the teams.

So the recorded synchronous collaborative activities constituted just a part of the process. Possible learning gains of the activity can not be simply attributed to the collaborative task, but prior (or intermediate) to the collaborative task cognitive elaboration during the phase of solving individually the problem cannot be neglected.

Peer tutoring behavior: One of the main objectives of the study has been to examine if there have been patterns of peer tutoring in the dyads. In most cases, there is an obvious discrepancy in the background knowledge of the partners in the beginning of the task. That is expected since the Greek students have not been taught anything about lists and recursion which is a core concept related to the given problem. On the other hand, concerning the pairs that developed flow-chart diagrams using Synergo, the German partners were not familiar neither with the tool, nor with flow-chart notation. Interestingly, many students seemed to be aware that this difference in prior knowledge could provide benefits for both partners. E.g. the dialogue logs between two collaborators included:

GR ": "so we both know different parts and we will connect them!!"

DE: "sounds good :)"

During the analysis phase we identified peer-tutoring instances between the two partners and gains of new knowledge.

In the following we study further this behavioral pattern, in relation to more specific learning objectives, relating to the use of lists and recursion.

Learning about lists and their usage: The first thing that was necessary for the students to comprehend in order to solve the given problem was lists and their properties. Greek students had limited prior teaching of lists, although they were quite familiar with arrays. On the contrary, teaching of lists was part of the course of the German students. Analysis showed that some students had worked on their own prior to collaborating with their partners, and seemed to have acquired a good knowledge of lists before the sessions started. However, in some cases Greek students initially had misconceptions of the properties of lists. They could assimilate the properties of lists to their previously existing understanding of arrays; they however had not accommodated their knowledge so that it would cover the main distinctive property of lists, i.e. the possibility to access, extract or add only one element at a time. The German partners explained the differences between lists and arrays to the Greeks, and the latter gained the provided knowledge with no significant difficulty. A characteristic example is provided below:

GR: "So now we have to compare a[1] and b[2]. So we add 1 to j and leave i as it was. Check it out."

DE: "But we have only the methods "stock", "first" and "rest" to get the elements of the list. You can't directly take the element 2 or other. You can only take the first element and then you must give out"

DE: "the rest. a = rest(a)"

DE: "You have an idea for handling Arrays, but not for the lists. Do you know what I mean?"

GR: "As far as I have understood, what we do is cut off the biggest element, replace the list that includes it with rest(list) and leave the other list untouched. It is true that I don't know how to deal with lists, I have never faced such problems."

In the first message the Greek partner tries to solve a problem involving lists using the properties and notations of arrays. However, the last message indicates knowledge and comprehension of the list properties. This contradiction becomes more obvious after a while, when the Greek student uses again array notation for representing list elements in the diagram. The depth of Greek students' comprehension of lists does not reach the extent so that she could *apply* this knowledge properly. Applying knowledge corresponds to a higher skill that indicates learning in more depth that just knowing something (Bloom (1956) The German student responds again:

DE: "You can't take b(i) and a(i), because it isn't a array. You can only put an element at the front of the list. ;-)"

After that, the Greek student corrected immediately the mistake with notations appropriate to lists. Such an extract indicates that collaboration can boost knowledge skills in terms of upgrading from "surface" knowledge to application of knowledge and even reflection on it. Similar episodes were reported in the activities of most pairs.

Learning about recursion: Learning about recursion in contrast to iteration (in the context of algorithms) by Greek students was the major challenge of the activity, since Greek students were not expected to know what recursion is. We expected to find effective peer-tutoring instances of German students helping Greek students learn what recursion is.

Although almost all German students mastered recursion, the dyads were diverse in terms of initial knowledge about recursion by Greek students and in terms of learning gains. About half of Greek students did not know anything about recursion. Some of them had some related knowledge that they could not associate with the term, and there were a few that had learnt about recursion prior to the collaborative sessions and could apply their knowledge in the development of the diagrams.

In most cases there was an imbalance of knowledge that we expected to be beneficial first for Greek partners, but for German partners as well, because explaining something helps learning it in more depth (Ploetzner et. al. 1999). In about half of the cases there were strong indications of such a success. For example while two partners had developed a first version of the diagram they exchanged the following messages:

1) DE: it is a recursion. and we have to transform it so that it works with loop

2) GR: could we write stock(first(y),merge(x,y))?

3) DE : no that would still be recursion. We can't use merge while we are in merge.

4) DE : shall I send you my code without recursion?

5) GR: but how are we supposed to use the operation stock if we HAVE to add the list name? I mean can we name the new list just "merge" or "list3"?

6) GR : it's the only way of preventing a recursion

7) DE : I think it's ok if we use a new list.

The Greek partner seems to understand what the German stated in message 3 “no that would still be recursion. We can't use merge while we are in merge”. Later on not only does he show understanding, but he provides a proposal for the right solution as well: “can we name the new list just "merge" or "list3"?”, “it's the only way of preventing a recursion”.

However, we noticed other cases that the fact that only the German student knew about recursion prevented true collaboration and learning gains. German students developed solutions almost on their own leaving their Greek partner observing the diagrams and contributing just trivial things. Even questioning about recursion did not necessarily lead to learning. An example is the following dialogue:

GR: “…but could you please explain what a recursive program is and what an iterative program is? Please?”

DE: “Ok, I try it. In German no problem, but in English. :-) Wait a moment, I must think about this. :-)”

GR: “Ok Take your time, it's OK :-)”

DE: “The algorithm in the tasks is a function. The name is "merge". In line 3 and 4 in stock() the algorithm calls itself. That is called "recursiv".”

GR: “Oh, I get it. I know what this is, I just knew only the Greek word for that, not the English...sorry.”

Although the Greek student claims to know what recursion is, his practice after this short dialogue proves the opposite. A possible explanation of this problem can be related to the means of assessment of the task which can highly influence the quality and orientation of learning (Knight, 1995). For the grading of the task, the final solution and the “level of collaboration” were taken into account, but the comprehension of recursion was not demanded in any test that the Greek students had to take in the context of the relevant course. So, they were not highly motivated to gain this knowledge and use it after the end of the collaborative activity.

Technical and usability problems: Some groups were faced with technical problems, so for instance some groups decided to switch tools. These technical problems appeared in different ways. Most of the problems were caused by network latency leaks in combination with usability issues due to the complexity of the tools.

For instance, one of the tools (FreeStyler) has many different workspace layers . The tool provides two types of layers, one for handwriting and drawing and the other for building more complex models. The graph layer supports simulation functionality (e.g petri nets) which may cause unexpected waiting in combination with network latency. For example while testing in fast network infrastructure, users "feel" immediately when they are working in the wrong layer or wrong mode and switch natively to the preferred mode. In the observed cases technical problems were attributed to a software limitation that was termed "a bug" by the students:

"GR: the bugs from hell just reappeared, wait a sec, I am going to restart freestyler
DE: for me it works just fine though
GR: okay, (I am) back"

Conclusions and Perspectives

In this paper we described organizational and technical affordances for long-distance collaborative activities across national borders. Based on our analyses we conclude that for complex learning scenarios, successful collaboration with advanced learning support environments is possible, yet requires careful preparation and planning of the support structures for the students. An important issue to consider also is the setup of the scenario in a joint pedagogical framework, such that all subjects involved benefit from the learning experience and feel motivated to participate actively, since the withdrawal of a partner cannot be compensated easily in a remote situation; creating personal bonds between the participants up-front is a means to avoid the loss of students. One important lesson learnt is the necessity to embed these learning scenarios into a meaningful and authentic context in the education of the institutions involved, especially in a multi-national or cross-cultural situation.

Both tools used have been proved very relevant for this activity. In addition the students used many communication tools and auxiliary material, often in the form of preliminary versions of the required solution in other representations, like in pseudo code. The fact that both tools used permit logging of student's activity, has been proven very useful for analysis of the actions and understanding the process. Modelling activities in the workspace have shown a wide variety in terms both of the number of actions and the distribution between the partners, as the overall number ranged between approx. 300 and 2500 workspace actions (creation, moving, modification and deletion of objects) and shares of initiative between 5:1 and 5:3 actions. It was found that there was no significant bias in the workspace actions towards either Greek or German students.

An interesting finding has been that from the analysis of the chat logs and the frequent episodes of peer tutoring in both ways we conclude that successful collaboration can even be established while the peers have different background knowledge. The turn taking example demonstrates such collaboration. We observed that – as intended by the design of the learning situation – Greek student gained a solid understanding of recursion and iteration, while German students intensified their skills in handling diagrammatic representations.

We believe that the findings of this study may be useful in setting up similar collaborative activities across boarders, that may lead to enforcement of educational co-operation and competency development, while exposing the students to an environment of multinational work and use of tools for distance collaborative activities, without requiring from them to leave their usual learning practice and environment. Considering this first study a useful experience, we plan to intensify our efforts towards this goal by establishing these joint activities on a regular basis.

This work was supported by the IKYDA program (DETAV-CSCA) and the European Research Team CAViCoLA in the Kaleidoscope Network of Excellence (Contract No. 507838).

References

Avouris N., Margaritis M., and Komis V., (2004). Modelling interaction during small-group synchronous problem-solving activities: The Synergo approach, 2nd Int. Workshop on Designing Computational Models of Collaborative Learning Interaction, 7th ITS2004, Maceio, Brasil, September 2004.

Bloom, B. S. (1956). Taxonomy of educational objectives, handbook 1: Cognitive domain. New York: Longmans Green.

Bologna Secretariat website (2006). http://www.dfes.gov.uk./bologna (last visited April 2006)

Clark, H.H. & Schaefer, E.F. (1989). Contributing to discourse. Cognitive Science, 13, 259-294.

Gaßner, K (2003). Diskussionen als Szenario zur Ko-Konstruktion von Wissen. Dissertation. Faculty of Engineering Sciences, University Duisburg-Essen.

Garrison, D. R., Anderson, T., & Archer, W. (2000). Critical Inquiry in a text-based environment: Computer Conferencing in higher education. The Internet and Higher Education 2 (2-3): 1-19.

Gunawardena, C N.., Lowe, C. A., Anderson, T. (1997). Analysis of global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing. J. of Educational Computer Research, 17(4), 397-431.

Knight, P. (1995). Assessment for learning in higher education. London: Kogan Page

Jansen, M (2003). MatchMaker - A Framework to Support Collaborative Java Applications, In U. Hoppe, F. Verdejo & J. Kay (eds.): Proceedings of the 11th Conference on Artificial Intelligence in Education, pp. 535-536. Amsterdam, IOS Press.

Lingnau, A., Kuhn, M., Harrer, A., Hofmann, D., Fendrich, M., Hoppe, H.U. (2003). Enriching Traditional Classroom Scenarios by Seamless Integration of Interactive Media. In V. Devedzic, J. Spector, D. Sampson and Kinshuk (Eds.). Advanced Learning Technologies: Technology Enhanced Learning, Los Alamitos, CA: IEEE Computer Society.

Murray, T., & Arroyo, I. (2002). Towards measuring and maintaining the zone of proximal development in adaptive instructional systems. In S.A. Cerri, et.al (ed), Intelligent Tutoring Systems, 6th ITS 2002, volume 2363 of LNCS, pages 749–758. Springer-Verlag, 2002.

Ploetzner R., Dillenbourg P., Praier M. & Traum D. (1999). Learning by explaining to oneself and to others. In P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches (pp. 103-121). Oxford: Elsevier

Smith, J. B. (1994). Collective Intelligence in Computer-Based Collaboration. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Vygotski, L.S., (1978). Mind in Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge, Massachsetts.