THE ARCHITECTURE AND EVALUATION OF A COLLABORATIVE LEARNING ENVIRONMENT

Meletis Margaritis, Nikolaos Avouris, Vassilis Komis

ABSTRACT
ModellingSpace is an open learning environment, particularly suitable for science education, which permits building of models by collaborating partners in various educational settings. This paper describes the main features of the ModellingSpace environment and in particular issues related with coordination and communication during problem solving. A number of evaluation studies of ModellingSpace, which have been recently contacted are also reported here. Through these studies the effectiveness and the limitations of the proposed architecture are identified and discussed.

KEYWORDS
Collaborative learning, computer-supported collaborative problem solving, synchronous collaborative software, groupware

INTRODUCTION

This paper describes the main aspects of the architecture of ModellingSpace, an open learning environment that supports real-time and asynchronous collaboration of small groups of students, engaged in problem solving. In particular the paper focuses on the coordination and communication mechanisms developed to support collaboration of ModellingSpace users. An evaluation study involving ModellingSpace, in the context of an authentic educational activity, is also discussed. ModellingSpace is particularly suitable for science education, since it promotes an exploratory modelling learning approach. ModellingSpace is an *open environment*, i.e. it permits authoring of new primitive entities, the models' building blocks, by educators or students. To these entities visual behaviour can be assigned, based on multimedia background material (images or video). Subsequently, the users of ModellingSpace can build and explore models based on inter-related primitive entities. There are various ways of inter-relating entities. Some relations are of qualitative nature representing vague verbal expressions (e.g. A affects B) or more precise expressions (e.g. A is proportional to B) or mathematical expressions (e.g. B= 5*A+3).

The entities and relations are manipulated by students collaboratively and this meaning-making activity involves testing of the behaviour of the individual entities in the frame of the constructed model, as well as selecting of alternative model representations. This activity can also involve text-based discussion of distant partners, through an instant-messaging tool, which permits interleaving of dialogue and action in the shared drawing board.

This paper describes the key design decisions of the ModellingSpace software and in particular issues related with control of interaction and dialogue, representation of the entities and models in a format that permits exchange of primitive material, and considerations for network bandwidth limitations. This environment has been designed and built, based on experience with existing previous tools, like *ModelsCreator 2.0* (Komis et al. 2001), which have been used in the past for teaching science and multi-disciplinary subjects in various educational settings, see Komis et al. 2002, Fidas et al. 2002b. Some findings of evaluation studies, involving ModellingSpace are also discussed in this paper. In these studies the effect of alternative coordination strategies and of heterogeneous sets of primitive entities on collaborative modelling activities, are investigated.

MODELLINGSPACE SYSTEM ARCHITECTURE

This section presents the main aspects of the architecture of the ModellingSpace (MS) system together with the main technological decisions concerning the system that has been developed. Special focus is provided on the communication and coordination mechanisms. Subsequently, evaluation studies of the system are presented. For a more detailed description of the design decisions that lead to this architecture, see Avouris et al. (2003b). The main decisions concerning the architecture are related to the development of the synchronous and asynchronous collaboration functionality, as well as the integration of the meta-cognitive analysis tools in the architecture. The decisions related to the architecture of the stand-alone modelling tools, i.e. the *Model editor* and *Entity editor*, are based in some extend on the existing ModelsCreator functionality and design (Komis et al. 2001, Dimitracopoulou et al. 1999).

Components of ModellingSpace architecture

The ModellingSpace (MS) software is a *client-server distributed application*, which comprises a suite of interconnected tools to support collaborative modelling activities. MS is an environment that supports individual and collaborative building of various kinds of models. It includes tools that permit building and editing of primitive entities, building and exploring models that are made of primitive entities, synchronous and asynchronous interaction of students, collocated or at a distance who collaborate in building models and tools that support analysis of modelling activities. The open character of MS means that students have access to an open set of primitive entities that can be used for building these models. A result of this characteristic is that the collaborating partners may reason using heterogeneous sets of primitive entities, in order to obtain a solution, as discussed in the following.

The main components of the architecture are therefore the following: (a) the Model Editor, Entity Editor, and the Analysis & supervision tools, residing in the client node and (b) the Common Repository and the Community support environment, to be found in the server node.

The main functionality of the Modelling environment is described through figure 1, which shows a typical model building activity.

Modellingspace main functionality

The primitive *entities*, i.e. the building blocks of the models, are stored in client local libraries, which can be shared by all users of the workstation (public libraries) or can be private libraries of the particular user. MS supports building of different kinds of models, like *semi-quantitative models*, i.e. models in which the entities are related by semi-quantitative relations (E1 is-proportional-to E2), hiding the underlying mathematical relation from the user, *quantitative models* in which the entities are related through quantitative mathematical relations, also models that can contain *graph-based relations*, and relations that are defined through *tables of values*. Combinations of these types of relations can co-exist in a model. Finally the users can define static relations, usually of verbal form, which cannot generate dynamic behaviour, like in the case of concept maps. This latter type of model is used in one of the two evaluation experiments discussed in this paper, while in an evaluation study reported by Fidas et al. (2002) a quantitative model has been used. On the left-hand side column of figure 1 a library of entities is shown, while on the right hand-side a library of available relations is shown. Alternative views can be produced by a model, in figure 1 a *graph*, on the right hand part (b) of the modelling space represents a relation between two entity properties. One can observe the similarity of this graph to the original graphical definition of the relation, shown in the left part (a) of the figure. Also data describing the activity can be produced in the form of logfiles and are available for inspection by the users or researchers. This latter representation is further discussed in the following, under the *analysis tools* perspective.

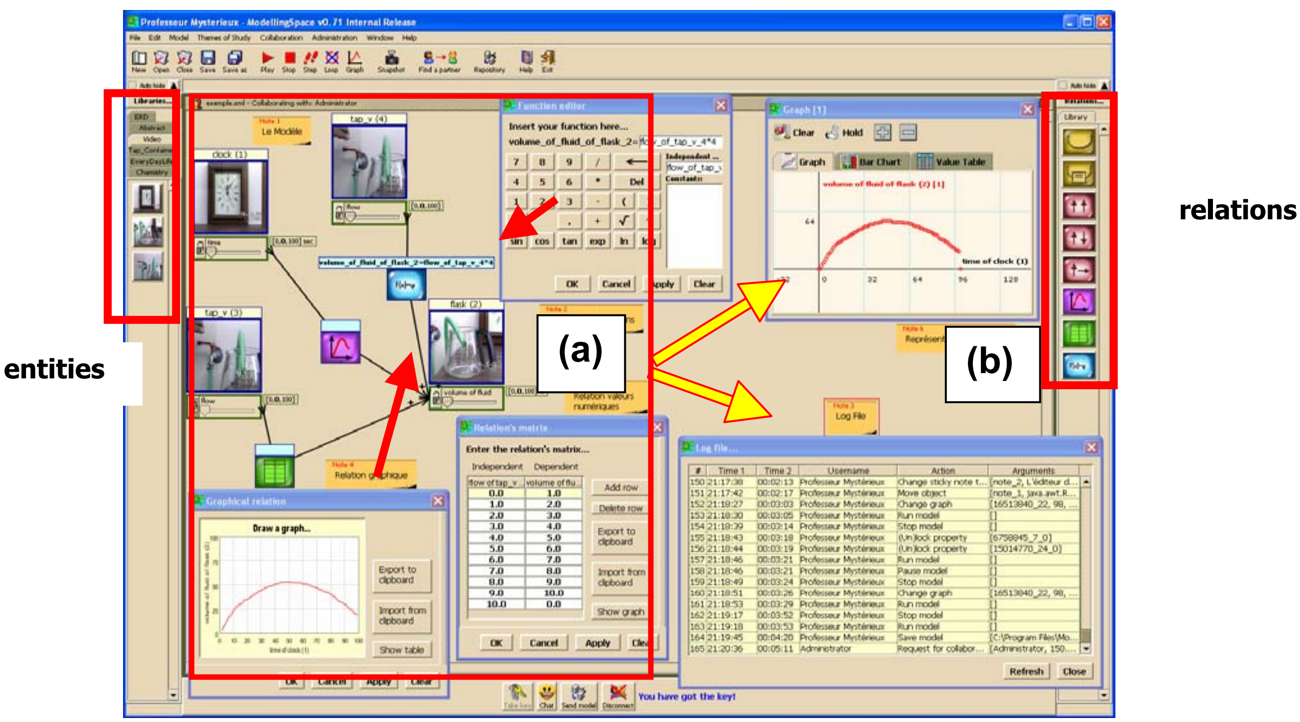

Figure 1. ModellingSpace model window : (a) Creation of a model containing multiple types of relations, (b) A graph view of a relation and an activity logfile

Communication and collaboration support

Synchronous and asynchronous collaboration for modelling is a case of computer supported collaboration based on the concept of shared artefact/ work surface (Dix et al, 1998). The related notion of *feed-through the artefact* implies that one participant's manipulation of shared objects can be observed by the other participants. This communication through the artefact can be as important as direct communication between participants, as observed in

(Avouris et al. 2003a and Komis et al. 2002).

Various architectural decisions are related to this framework. Considering that the collaborative activity will be done mainly between partners at a distance, the direct communication mechanism has to be defined. A text communication has been used in this case, since other media like video and audio are considered not suitable as discussed in Avouris et al. (2003b). One additional decision is related to the design of the shared activity space, also discussed in the following. In ModellingSpace a mixture of alternatives is provided. A strict WYSIWIS (what you see is what I see) is allowed in the main model-editing window. We believe that activity in this area should be faithfully reproduced in all participants' workstations. This is because most of communication and reasoning is based on this shared viewpoint, which becomes the main grounding mechanism of dialogue and through which eventually common understanding can occur. Deviation from this, results in confusion of partners since misunderstandings can be generated due to different views when partners are allowed to scroll to different viewpoints, while no strong coupling of the shared view and the direct communication can be achieved. However all additional operations outside this shared workspace, e.g. relating to browsing of the themes of study activity sheets and other auxiliary material, saving of the model and running graph tools with alternative representations of the built model, should be performed independently by partners involved, a *model level coupling approach* according to Suthers (2001).

Coordination support

One other important decision is related to the design of a coordination mechanism for the activity in the shared workspace. In the case of ModellingSpace we have built a floor control mechanism. This decision is based on studies that have demonstrated that such mechanisms instead of creating confusion, seem to promote collaborative activity, as discussed in the study reported in this paper. The proposed coordination mechanism involves the notion of the *Action Enabling Key*, which is owned by one of the participants at any given time. This key owner can then act in the shared workspace, while the rest just observe this activity and make comments through the chat tool. This mechanism should be supported by *key request, key accept, key reject* functions. Experiments with this floor control mechanism, see also (Fidas et al. 2001) and (Komis et al. 2002), demonstrate that it improves reasoning about action, as partners need to reason and negotiate during key requests. In figure 2, the user interface of the key control functionality is shown

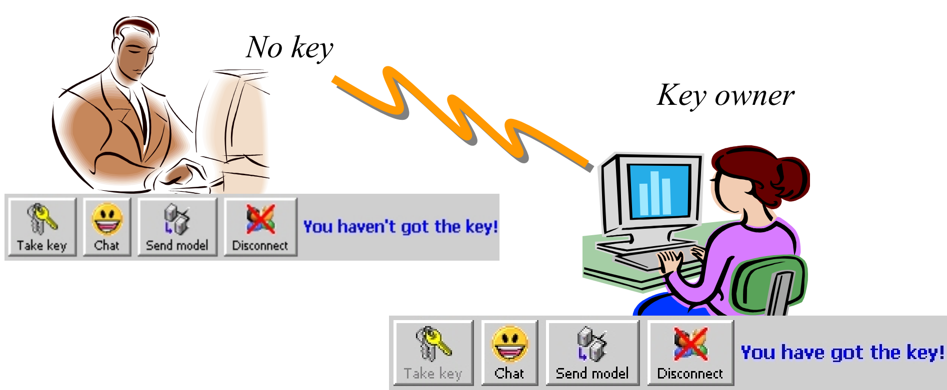

Figure 2. Key passing coordination mechanism

Design of the shared drawing board
The main activity space of the MS modelling environment needs to be shared by multiple actors, permitting collaborative modelling activities of learning actors at a distance. Sharing this activity space is achieved using a peer-to-peer connection between two or more client nodes. Through this connection, the necessary control messages are exchanged, which permit the WYSIWIS shared drawing board effect. This approach is preferred to a server-based architecture, since the latter would have been a bottleneck in case many groups were simultaneously engaged in collaboration. The size of the groups engaged in synchronous collaboration is expected to be small, so point-to-point connection is feasible. As described in Avouris et al. (2003b), the exchanged messages are going to be of small size, as due to replication the only information exchanged relates to control of modelling activities (e.g. add entity E_i to the (x,y)), while the entity E_i itself is not usually transferred between the distant nodes.
In case that a primitive entity is used by one of the partners and cannot be found in peers' workstations during modelling, a need arises to transmit this entity, possibly with large multimedia files, to collaborating peers in order to synchronize the peer applications. This can create serious disruption in the smooth collaboration to all collaborating partners, due to long download times. A solution proposed for this problem is to send only light control messages to the peers (chat and change of state), including the structure of new primitive entities, while the heavy multimedia files associated to these entities, if required, to be send through the server to the requesting peers, without creating disruption to the rest of the group.

Communication tool
In the frame of the collaborative use of MS, a dialogue tool has been integrated, which is based on an instant messaging protocol, using the same point-to-point connection and protocol discussed in the context of the shared activity space. Through this, text messages are exchanged during collaborative problem solving.

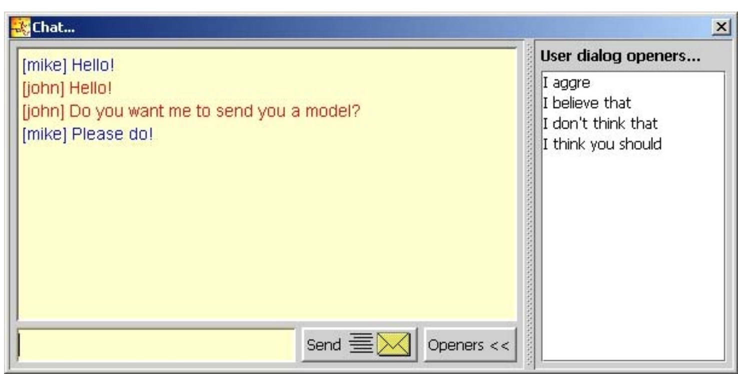

Figure 3. The structure chat window

The chat tool, which is activated from the collaboration panel, is equipped with dialogue openers. This way the user can select the opening phrase of the utterance and thus classify indirectly the speech act. There is a lot of controversy associated with structured dialogue mechanisms. Some researchers believe that they interfere with interaction and should be

avoided, while others believe that they support development of meta-cognitive skills and in addition they facilitate analysis of communication and collaboration.
In MS we have opted for a parametric approach, where the researcher is able to decide on the dialogue openers if they need to be introduced. So an appropriate dialogue option in the administration menu has been introduced through which the dialogue openers are defined, as shown in figure 3.

Deictic interaction
Other means for exchange of text messages are the sticky notes (text containers positioned in the activity space) which are treated, in terms of the architecture, as special kind of entities, with internal properties: *owner, time of creation, text_content*. Through the sticky notes, gestures can be simulated, since a sticky note inserted in the drawing board can be related to an object in the shared activity space and through this synchronous or asynchronous collaboration can be achieved. An example of a sticky note is shown in figure 4.

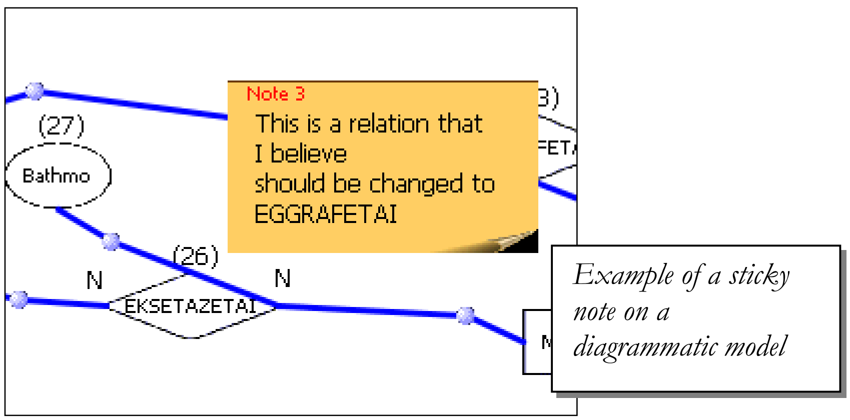

Figure 4. A sticky note has been inserted in a model, referring to an entity that makes part of a model

Supervision and monitoring of problem solving
The MS is designed to be a user-sensitive environment, providing different functionality to different actors. So the teacher can use the tool for supervising simultaneously many groups of students, and share many collaboration windows, while special permission are allocated to them in relation to coordination of collaboration. An example of use of this functionality is discussed in the evaluation study of this paper, see figure 7.

Analysis Tools
An additional feature of MS relates to analysis of modelling and collaboration activities. So a set of Analysis tools is included in the environment. These are mainly planned to be used by the teachers and researchers, while limitted versions of the tools can be used in some cases by students as meta-cognitive aids. For instance the student tools present the model and permit playback of the modelling activity while problem solving is in progress. The main functionality of the Analysis tool is the presentation and processing of logfiles which have been produced during MS activities. These logfiles contain actions and text messages of all partners engaged in modelling, in sequential order. An example of a logfile of a developed prototype is shown in figure 5. The logfile is based on the format of the exchanged control and chat messages and can be stored in XML form. This file can be viewed, commended and annotated by a researcher using an adequate analysis framework, as discussed by Avouris et

al. (2003a, 2003c). A related functionality of the analysis tool is its capability of posterior reproduction of the modelling activity, using the logfile, in a step-by-step or continuous way. This is complementary to the logfile inspection and annotation functionality. The activity can be reproduced using the *playback tool*. Annotation though this playback tool can also be done as discussed in more detail in Avouris et al. (2003c).

```
1) 2:22:01 μμ  00 : 48 : 55   1       Request Key
2) 2:22:11 μμ  00 : 49 : 05   2       Accept To Give The Key
3) 2:22:18 μμ  00 : 49 : 12   1       Chat   I asked for the key
4) 2:22:26 μμ  00 : 49 : 20   1       Chat   oκ I got it
5) 2:22:32 μμ  00 : 49 : 26   1       Rename Object  Ellipse  1  from  END  USER  toEND  USER  #2
(A2412)
6) 2:24:12 μμ  00 : 51 : 06   1       Chat   Get  the  key  and  change  all  the  relations  with
those we have connected the LANS
7) 2:25:11 μμ  00 : 52 : 05   2       Chat   ok
8) 2:25:14 μμ  00 : 52 : 08   2       Request Key
9) 2:25:19 μμ  00 : 52 : 13   1       Accept To Give The Key
```

Figure 5. Extract of a logfile from collaborative modelling (from Fidas et al. 2002)

The annotated or original logfiles are in pure XML form, i.e. they do not contain the multimedia entity files involved in the developed model, but instead they contain references to them by their unique identifier GUID. So if an entity X is used by a logfile L and is not available in the local libraries, the analyst needs to search and download the related entities in order to be able to playback the model and reproduce the activity. In case of missing entities the environment will reproduce them by a default entity with no behaviour or iconic representation associated. This decision to disentangle, the logfiles from the heavy structures associated with entities is made in order to keep the logfiles small in size and facilitate their easy exchange and storing. It is assumed that a researcher who is involved in analysis of a modelling activity has access to the primitive entities used. The logfiles can be stored and exchanged in various formats including XML and the tools are based on a database of logfiles, which serve for studies of modelling activities.

Finally, since many ethnographic studies are expected to be performed during the case studies involving the MS environment, a tool for analysis of video and audio recordings and synchronisation of such behavioural data files with activity logfiles has also been built. This tool is integrated in the analysis environment. It is able to playback various formats of video and audio and associate points in video/audio to events and comments in the logfiles.

EVALUATION STUDIES

The environment discussed in the previous section has been recently used in a number of evaluation studies. The first case concerns a computer science University course. The collaborative problem solving experiment took place in the frame of the Laboratory of the Undergraduate course “Data and Knowledge Based Systems” of the Electrical & Computer Engineering Department of the University of Patras, during the winter 2002 semester. The objective of this experiment was to evaluate the effectiveness of the architecture in supporting distance and local collaborative modelling activities and to evaluate the impact of alternative design approaches, according to the rationale discussed in the previous section, on collaboration and performance. In particular a study of the effect of the key passing coordination mechanism took place, as discussed in the following.

Twenty-two (22) students participated in the experiment in the frame of a scheduled laboratory session. Eleven (11) pairs of students with similar characteristics were formed, that collaborated. The collaborating groups, dispersed in the computer lab, interacted for a certain period of time, using MS in order to tackle a given data-modelling problem. The location of the group members was such within the lab that they could interact exclusively through the provided tools, thus simulating distance problem-solving conditions. In figure 6 the lab during the experiment is shown. The tutor did not intervene during the problem solving process, except in the final stage of the lab, when an overview of some of the solutions was displayed and discussed, as shown in figure 7.

Figure 6. Setting of Modellingspace use in a data modelling lab

Each pair of students was asked to produce, at the end of the laboratory session, a single solution to the problem, using the collaborative problem-solving environment MS. The task was to build an ERD (entity-relation diagram) representing the mircorworld of a University Department. Extracts of typical solutions, produced during the process are shown in figure 7.

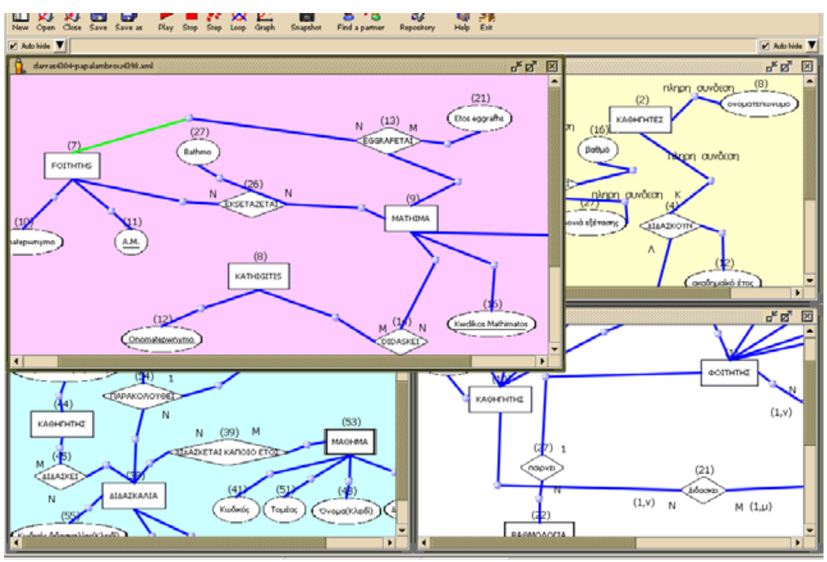

Figure 7. Real-time overview of multiple student solutions in the tutor workstation

Five pairs of the students (group A) had no explicit coordination mechanism imposed to them, while six pairs used the key-passing mechanism described in the previous section (group B). All groups produced acceptable solutions to the given problem within the allocated time. The mean performance of the two groups were very similar (group A, 7.6 and group B 7.7). It should also be observed that both groups performed in comparable way in a pre-test subject matter evaluation (group A mean score = 7.6 and group B mean score 7.2). So a first observation was that the existence or lack of the coordination mechanism did not seem to affect the performance of the students.

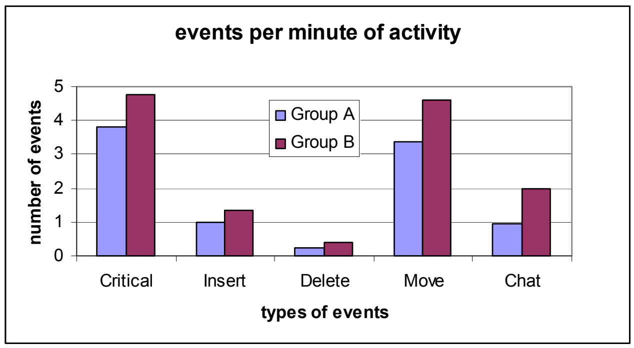

Figure 8. Events per minute of activity for groups A and B

By studying in more detail the collaboration activity through the produced logfiles and solutions, it was observed that group B was more active than group A, as shown in figure 8.
This observation was valid for all types of events (insertion of objects, deletion of objects, moving objects and exchange of chat messages). The most significant difference was in the last type of events. The complexity of the produced solutions by group B was also higher, while this did not mean that the quality of solutions was necessarily higher. This finding was not expected, as it was believed that continuous possession of the activity key by both partners, would result in higher overall activity.

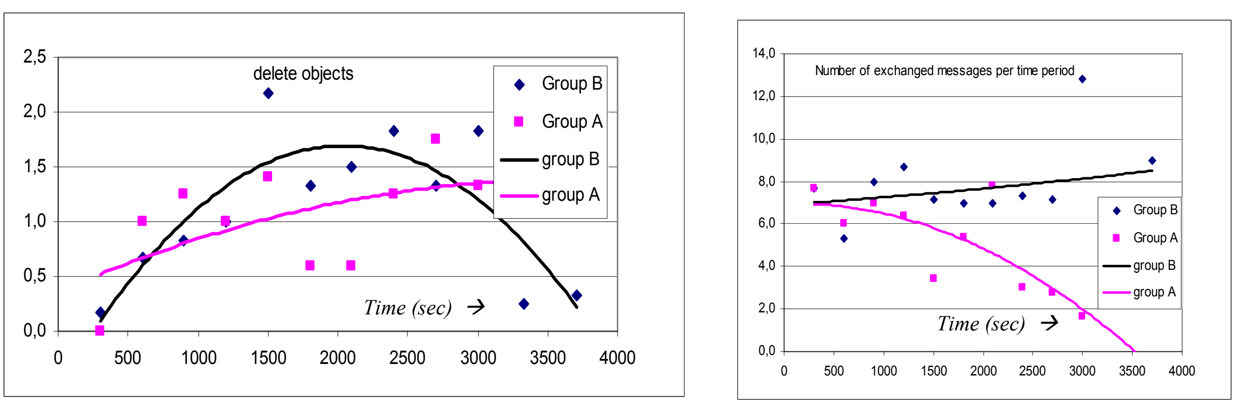

Figure 9. Frequency of events in time slots of activity (delete and chat events)

A more detailed analysis of the occurrence of various types of events during problem solving activity also took place. The frequency of occurrence of events per time interval was measured and plotted in the graphs of figure 9. The graphs of the *delete object* and *chat* events are shown in this figure. A second order polynomial interpolation of the points of the two groups has been included in the graphs, in order to depict more clearly the trends. From fig. 9, it is evident that the key-passing coordination mechanism that was used in the case of group B, has affected the behaviour of the relevant pairs of students.

The *deletion of object* events in group B seem to fade towards the end of the activity, while in group A, there is a constant increase of these events during the problem solving activity, since the possession of the key permits a reactive behaviour by the partners.

As far as the *exchanged chat messages*, it seems that group B maintains a constant level of direct communication activity throughout problem solving, since the key-passing mechanism, obliges one of the two partners to take the passive role, and therefore use the chat tool as a means of expression, while in group A, exchange of textual messages decreases with time, as the partners probably prefer to use direct manipulation instead of verbal expression of arguments as the problem solution is developed.

As a conclusion, the explicit coordination mechanism used in this evaluation study did not seem to inhibit the problem solving process or affect the quality of solutions. It contrary, it seems that by imposing this mechanism, the students were forced to exchange more chat messages and to act more in the common activity place, thus increasing the degree of collaboration within each group.

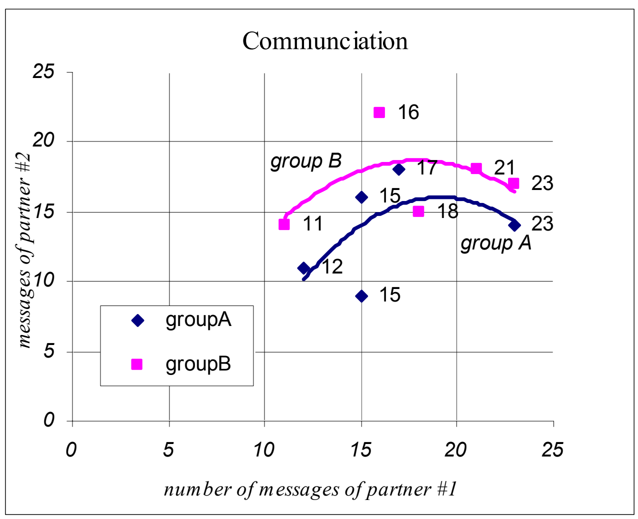

Figure 10. Communication in terms of exchanged text messages for group A (shared entities) and group B (heterogeneous entities)

This finding seems to be of similar nature to another recent study by Fidas et al. (2002), in which, using a similar collaborative modelling environment, the effect of heterogeneity of the primitive entities in problem solving was studied. In this study two groups of secondary education students were formed, that differed only in terms of the primitive entities available to each pair, out of which they were asked to build a model in a collaborative way. In one group both partners of each pair shared the same primitive entities, while in the second group some of the necessary entities were owned by only one of the two partners. The second group was overall more active in terms of actions and dialogue, see figure 10. Furthermore, by studying the history of the solution components it was found that in the second group, considerably more components were owned by both partners (69%) than in the first group (53%). It therefore seemed that the heterogeneity of entities instead of creating additional difficulty to collaborating partners, as originally expected, it was a reason for more involvement and deeper discussions, without any deterioration of the quality of the produced solutions.

CONCLUSIONS

In this paper we outlined the main aspects of a new collaborative modelling environment and discussed some preliminary findings that were produced from its experimental use in a number of evaluation studies. The collaboration-support features of ModellingSpace, the capacity of building diverse models of qualitative, quantitative or descriptive nature, the open character of the environment in terms of primitive entities to be used as building blocks for modelling activity, are some of the main features of the environment. Also the integration of analysis tools in the ModellingSpace environment make it a particularly suitable tool for experimentation on collaborative learning and science education for various educational levels. The design of the MS environment is based on a parametric nature of many features, like the dialogue openers and the coordination mechanisms to be used. This way investigation of the effectiveness of these parameters in collaboration and learning can be performed. Today there are still many issues relating to collaborative learning in science, that necessitate further research (Stahl 2002). Often the most intuitive design of an educational environment might not be the most effective solution in terms of educational result, as the findings of the studies reported here have demonstrated. So experimental tools are needed to support such studies. For this reason the ModellingSpace testbed, which is going soon to become available to the research community for experimental purposes (see www.modellingspace.net for details), can be an invaluable means towards the direction of better understanding of the related issues.

ACKNOWLEDGEMENT
The reported work has been performed in the frame of the IST-School of Tomorrow Project IST-2000-25385 “ModellingSpace”. In this project the partners are: the University of the Aegean, (GR), the University of Patras (GR), the University of Mons-Hainaut (B), the New University of Lisbon (PT), the University of Angers (F) and SchlumbergerSema (SP).

REFERENCES

1. Avouris N.M., Dimitracopoulou A., Komis V., (2003a), On analysis of collaborative problem solving: An object-oriented approach, J. of Human Behavior (in print)
2. Avouris N., Margaritis M., Komis V., Melendez R., Saez A., (2003b) ModellingSpace: Interaction Design and Architecture of a collaborative modelling environment, CBLIS, Nicosia 2003.
3. Avouris N., Komis V., Margaritis M., Fiotakis G., Tselios N. (2003c), Tools for Interaction and Collaboration Analysis of learning activities, CBLIS 2003, Nicosia 2003.
4. Dimitracopoulou A., Komis V. Apostolopoulos P. & Politis P. (1999). Design Principles of a New Modelling Environment Supporting Various Types of Reasoning

and Interdisciplinary Approaches, in Proc. of 9th Int. Conference of Artificial Intelligence in Education, IOS Press, Ohmsha, pp. 109-120.
5. Dix A., Finlay J., Abowd G, Beale R., (1998), Human-Computer Interaction, 2nd Edition, Prentice Hall
6. Fidas C., Komis V., Avouris N.M. (2001). Design of collaboration-support tools for group problem solving, Proceedings PC HCI 2001, pp. 263-268, Typorama Pub., December 2001, Patras, Greece.
7. Fidas C., Komis V., Avouris N.M., Dimitracopoulou A., (2002a), Collaborative Problem solving using an Open Modelling Environment, Proc. CSCL 2002, pp. 654-656, Erlbaum Assooc, Hillsdale NJ, 2002.
8. Fidas C., Komis V., Tzanavaris S., Avouris N., (2002b), Heterogeneity of learning material in synchronous computer-supported collaborative modelling, Computers and Education (submitted)
9. Komis V., Avouris N., Fidas C., (2002), Computer-supported collaborative concept mapping: Study of synchronous peer interaction, Education and Information Technologies vol.7, 2, pp.169-188, 2002
10. Komis V., Dimitracopoulou A., Politis P., Avouris N. (2001). Expérimentations exploratoires sur l'utilisation d'un environnement informatique de modélisation par petits groupes d'élèves, Sciences et Techniques Educatives, Vol. 8, no 1-2, pp.75-86.
11. Preece J, Rogers Y, Sharp H., (2002), Interaction Design beyond human-computer interaction, Willey & sons.
12. Stahl G. (ed), (2002), Proceedings Computer Supported Collaborative Learning 2002, Boulder Colorado, January 2002, Erlbaum Assoc. Hillsdale, NJ.
13. Suthers, D.D., (2001), Architectures for Computer Supported Collaborative Learning, Proc. IEEE int. Conf. on Advanced Learning Technologies, ICALT 2001, Madison, Wisconsin

N. Avouris, M. Margaritis, V. Komis, University of Patras, Patras, Greece
{N.Avouris, Margaritis} @ ee.upatras.gr, komis@upatras.gr