ModellingSpace: A tool for synchronous collaborative problem solving

Nikolaos Avouris, Vassilis Komis, Meletis Margaritis, Christos Fidas
University of Patras, GR-265 Rio Patras, Greece
N.Avouris@ee.upatras.gr, Komis@upatras.gr, Margaritis@ee.upatras.gr, Fidas@ee.upatras.gr

Abstract:
ModellingSpace is a learning environment that allows collaboration of partners, collocated or at a distance in various educational settings. This paper describes the main features of the architecture of the ModellingSpace environment and in particular issues related with coordination and communication during problem solving. The results of an evaluation study of ModellingSpace, which has been recently contacted in order to examine the effect of various levels of locking of objects in the shared activity space are also reported here. Through this study, the effectiveness and the limitations of the proposed architecture are identified and discussed.

Introduction

This paper describes the main aspects of the architecture of ModellingSpace, an open learning environment that supports synchronous 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 promotes an exploratory modelling learning approach. 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), see Dimitracopoulou and Komis (2003) for details on the reasoning approaches supported for modeling activities in this environment. 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 for teaching science and multi-disciplinary subjects (Komis et al. 2002, Fidas et al. 2004).
The typical context of use of this environment is in the frame of a school lab, where collocated students form groups and collaborate using ModellingSpace, in subjects like physics, maths, biology, computer technology or environmental studies subjects. Both secondary education and college level students have used this software environment in experimental basis. The modeling activity involves testing of the behaviour of the individual entities in the frame of the constructed model, as well as using alternative model representations, like graphs depicting the relation of model entities. Support is provided also for text-based communication of partners, through an integrated 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 locking of objects in the shared activity space and subsequently 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. Some findings of an evaluation study, involving ModellingSpace are also discussed in this paper.

ModellignSpace 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. Special focus is provided on the communication and coordination mechanisms. Subsequently, an evaluation study of the system is presented. For a more detailed description of the design decisions that lead to this architecture, see Avouris et al. (2003c). The main decisions concerning the architecture are related to the development of the synchronous and asynchronous collaboration functionality, as well as the integration of supervision tools in the architecture.

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. ModellingSpace has been built using an Abstract Collaborative Application Building Framework (ACABF) that is based on a model-level replication of the collaborating peer workstations. According to Dewan (1999) synchronous collaboration systems are based on some degree of replication in peer nodes. Systems like Microsoft NetMeeting and Shared X replicate the screen or window layers allowing single user applications to be simultaneously viewed by collaborating partners. In our case, we support replication of the model, so that rendering of the new state of the user environment is done using local resources. This way the exchanged messages are smaller and the response time is much improved, since user actions are replicated locally. The ACABF architecture defines an XML syntax for exchanged messages (messages involving coordination, shared activity space events and direct communication events). The ACABF architecture has already been used for building a number of synchronous collaboration support applications, like *ModelsCreator3*, *Representation2* and *Synergo*.

The main components of the ModellingSpace application are the following: (a) the Modelling Environment, the Entity Editor, and the Analysis & Supervision tools, residing in the client nodes and (b) the Common Repository and the Community support environment, as well as the Communication Relay Server 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. 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. Also the chat tool and the collaboration panel through which the action enabling key is exchanged are shown, the are further discussed in the following.

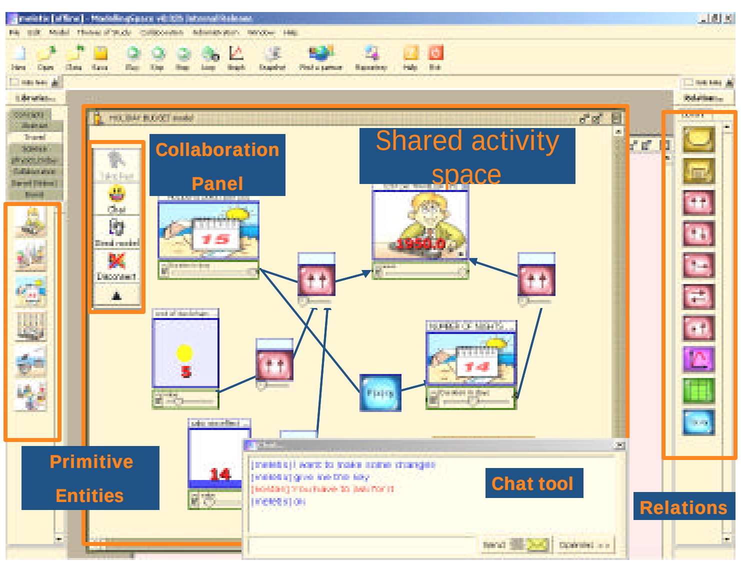

Figure 1: ModellingSpace model view

Communication and collaboration support

Synchronous collaboration for modelling is a case of computer supported collaboration based on the concept of shared artifact/ work surface (Dix et al, 1998), in which a graphical representation of the model is included. The related notion of feed-through the artifact implies that one participant's manipulation of shared objects can be observed by the other participants.

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 been defined based on text communication. Other media like video and audio are considered not suitable as discussed in Avouris et al. (2003c), since they impose a number of technical restrictions. 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-building 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, 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, which involves locking at various levels of the objects in the activity space. This decision is necessary in order to avoid users concurrently interacting with the same object, canceling each-others actions, like modifying the text of a concept or the value of variable. Alternative levels of locking may be imposed, subject to the users' or the teacher decision. As discussed in the following section of the paper, the level of locking affects users behaviour.

One alternative has been to impose a *model-level locking* mechanism. This involves the notion of the *Action Enabling Key*, a token which is owned by one of the participants at any given time, in which case the modeling window is locked for the other partners. 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 is supported by key request, key accept, key reject functions. Experiments with this floor control mechanism, see also (Fidas et al. 2001) and (Avouris et al. 2003b), demonstrate that it improves reasoning about action, as partners need to reason and negotiate during key requests, as discussed in the final section of this paper.

The second alternative is to impose an *object-level locking*. This is done transparently from the user, so no explicit floor control mechanism is imposed, like in the previous case and activity in the shared activity space is more fluent. However when a user selects an object in order to perform an operation, this is locked temporarily until the operation is terminated by this user. If other users in the mean time attempt an operation on the same object, their operations have no effect.

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. (2003c), the exchanged messages are 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, a *model-level replication* as discussed in the previous section.

The synchronization of objects in the shared activity space is a process that involves design decisions that permit not just accurate reproduction of the final state of the model after release of an object lock, but also accurate reproduction of the activity that took place during lock hold. An example is described in the following. Let us assume that a user is dragging the variable slider of the object *Timer*, shown in figure 2, back and forth for a while, in order to observe the effect of this variable on the variable *Temperature* through a graph. When the user stops dragging the slider, a message will be sent to the peer workstations which will include three values of the variable *Time*, these are the Maximum, Minimum and Final value of the variable, which will be used by the peer MS environments for reproducing accurately the same graph, as well as to bring the models in the same final state.

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. In order to tackle this problem we 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 that may be associated to these entities, are send through the Server

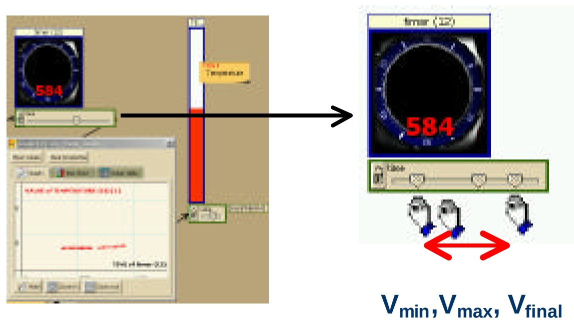

Figure 2. Dragging operation on an object value results on transmission of a synchronization message to peer workstations which contains Max, Min and Final values of Variable V

Supervision and Analysis Tools

An additional feature of MS relates to supervision of modelling and collaboration activities. So a set of supervision tools is included in the environment. These are mainly used by the teachers and researchers, while limited versions of the tools may 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 Supervision tool is the presentation and processing of logfiles which have been produced during MS activities. These logfiles contain actions and exchanged messages of all partners engaged in modelling, in sequential order. An example of a history logfile of a developed prototype is shown in figure 3. 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, 2004). A related functionality of the supervision tool is its capability of posterior reproduction of the modelling activity, using the logfile, in a step-by-step or continuous way. 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. (2004).

1) 00 : 48 : 55	U1	Request Key	
2) 00 : 49 : 05	U2	Accept To Give The Key	
3) 00 : 49 : 12	U1	Chat	I asked for the key
4) 00 : 49 : 20	U1	Chat	ok I got it
5) 00 : 49 : 26	U1	Rename Object	Ellipse 1 from END USER to END USER #2 (A2412)
6) 00 : 51 : 06	U1	Chat	Get the key and change all the relations with those connected t? the LANS
7) 00 : 52 : 05	U2	Chat	? ?
8) 00 : 52 : 08	U2	Request Key	
9) 00 : 52 : 13	U1	Accept To Give The Key	

Figure 3: Extract of a history logfile from collaborative modelling (from Fidas et al. 2002)

The annotated or original history logfiles are in 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 multimedia entities is made in order to keep the history logfiles small in size and facilitate their easy exchange and storing. 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. (Avouris et al. 2004), that can interact with the supervision environment.

Evaluation Study

The MS environment has been used in a number of evaluation studies. One typical study concerned a computer science University course. A number of 1st year students of the *Introduction in Computer Science* course, were requested to built collaboratively models (concept maps) describing the Internet. Examples of suggested concepts were *web client*, *web server*, *browser*, *IP*, etc. The objective of this experiment was to evaluate the effectiveness of the architecture in supporting distance collaborative modelling activities and to examine the impact of alternative protocols of locking of the common work surface in which the model is built.

Thirty-two (32) students participated in the experiment in the frame of a scheduled lab session. Sixteen (16) pairs of students with similar characteristics were formed. The collaborating pairs, dispersed in the computer lab, interacted for about 30', using ModellignSpace. The location of the partners was such within the class that they could interact exclusively through the provided tools. The tutor did not intervene during the problem solving process, except for facilitating use of the tools.

Each pair of students was asked to produce, by the end of the session, a single solution to the problem, using the collaborative problem-solving environment. Eight pairs of students (group A) used the key-passing floor control mechanism described in the previous section which imposes a model-level locking of the activity space, while the other eight pairs had no explicit floor control mechanism imposed on them (group B), thus using an object level locking protocol.

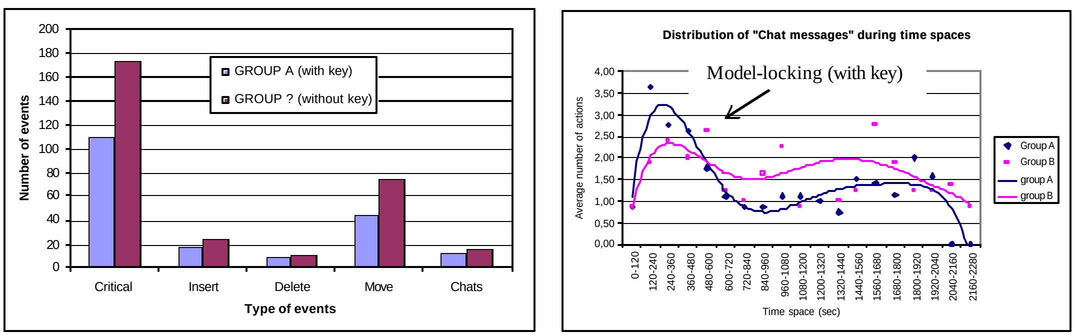

Figure 4. Comparison of behaviour of the two groups of students using different level locking: Group A: Model-level (with key) and Group B: Object level (without key)

In figure 4 a comparison of performance of the two groups is shown. On the left, the number of events in the activity space is shown. From this chart it is evident that group B was more active than A for all major categories of events. On the right the evolution of the exchanged chat messages is shown for the two groups during problem solving. The patterns of activity were very similar in both groups, as shown in fig.4. Overall, it was found that explicit floor control of the shared work surface did not inhibit problem solving. The solutions produced by the group of students who had to use explicit floor control were of similar quality to the reference group who did not use such a mechanism. Group B was more active than A, while interaction was more symmetrical among partners in the explicit floor control group A. On the other hand it should be observed that the lack of floor control did not seem to create any confusion or conflicts to the corresponding pairs of group B. However the size of the problem-solving groups (two partners) perhaps was too small to generalise on this observation, which might not hold for groups of more students. A more general conclusion of this study seems to be that by imposing explicit coordination mechanisms, the students have been obliged to negotiate on possession of the activity enabling key and thus argue at the meta-cognitive level of the activity and externalise their strategies, a fact that helps them deepen their collaboration, and lead to improved learning.

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 during an evaluation study. 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 for locking objects in the activity space. 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, 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 ModellingSpace, can be an invaluable means towards the direction of better understanding of the related issues.

Acknowledgements

This work has been performed in the frame of IST-2000-25385 “ModellingSpace” project (www.modellingspace.net).

References

Avouris N.M., Dimitracopoulou A., Komis V., (2003a) On analysis of collaborative problem solving: An object-oriented approach, *Computers in Human Behavior*, Vol. 19, Issue 2, March 2003, pp. 147-167.

Avouris N., M. Margaritis, V. Komis, (2003b), Real-Time Collaborative Problem Solving: A Study on Alternative Coordination Mechanisms, *Proc. of 3rd IEEE Intern. Conf. on Advanced Learning Technology (ICALT)*, pp.86-90, Athens, July 2003

Avouris N., Margaritis M., Komis V., Melendez R., Saez A., (2003c) ModellingSpace: Interaction Design and Architecture of a collaborative modelling environment, *Proc. CBLIS*, Nicosia 2003.

Avouris N., V. Komis, M. Margaritis, G. Fiotakis, An environment for studying collaborative learning activities, *Journal of International Forum of Educational Technology & Society*, Special Issue on Technology – Enhanced Learning, 7 (2), pp. 34-41, 2004 .

Dewan P., (1999), Architectures for collaborative applications, in Beaudouin-Lafon M. (ed), *Computer Supported Cooperative Work*, Wiley, Chichester, pp. 169-193.

Dimitracopoulou, A. Komis V. (2003), Design Principles for an open and wide ModellingSpace of modeling, collaboration and learning, *Proc. CBLIS 2003*, Nicosia 2003.

Dix A., Finlay J., Abowd G, Beale R., (1998), *Human-Computer Interaction*, 2nd Edition, Prentice Hall

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.

Fidas C., Komis V., Avouris N.M., Dimitracopoulou A., (2002), Collaborative Problem solving using an Open Modelling Environment, *Proc. CSCL 2002*, pp. 654-656, Erlbaum Assooc, Hillsdale NJ, 2002.

Fidas C., Komis V., Tzanavaris S., Avouris N., (2004), Heterogeneity of learning material in synchronous computer-supported collaborative modelling, *Computers and Education* (forthcoming).

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

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.

Stahl G. (ed), (2002), *Proceedings Computer Supported Collaborative Learning 2002*, Boulder Colorado, January 2002, Erlbaum Assoc. Hillsdale, NJ.

Suthers, D.D., (2001), Architectures for Computer Supported Collaborative Learning, *Proc. IEEE int. Conf. on Advanced Learning Technologies, ICALT 2001*, Madison, Wisconsin