Real-time Peer Collaboration in Open and Distance Learning

Nikolaos M. Avouris, Meletis Margaritis and Vassilis Komis

Abstract-- In this paper an overview of some key issues of computer-support for real-time collaborative problem solving in the context of Open and Distance Learning (ODL) is attempted. Aspects of the synchronous collaboration-support environment *Modelling Space* are described. This environment has been recently used in a number of empirical studies involving collaborative problem solving. From these studies it has been found that the representational and operation tools of the ModellingSpace common activity space influence considerably interaction and communication. Also the coordination mechanism used was found that affects problem solving and collaboration. This environment is proposed here as a tool that can be effectively integrated as a service to the students of ODL courses, in addition to more traditional asynchronous collaboration tools, improving community building.

Index Terms-- computer-supported collaborative learning, human-computer interaction, peer-to-peer computing, open and distance learning

I. Introduction

SYNCHRONOUS collaboration of students in the context of distance learning courses is very difficult for a number of technical and organizational reasons. For many practitioners of the field, synchronous collaboration is considered equal to video-mediated teleconferencing (e.g. Bouras et al. [5], Kato et al. [10]). This approach however necessitates high bandwidth connections and special equipment, not widely available to the students of distance learning courses, and for this reason, it is not widely used.

New technological advances in peer-to-peer (p2p) computing propose an alternative approach, which, as argued in this paper, is feasible to be implemented and used with the current commonly available infrastructure. P2p applications that facilitate file exchange have proliferated recently, while many other applications of this computing approach have been proposed, among which real-time collaboration has received special prominence (Lethin, [13]). So an alternative approach for synchronous collaboration is the use of low-bandwidth text-based communication facilities implemented over peer-to-peer interaction protocols. According to Lethin [13], the technical advantages of such an approach are related to fault tolerance, performance, and security, while as a result of the potential of powerful communication technologies in distributed form, new person-to-person interaction structures may emerge (Lopez and Skarmeta [14]). However, development of an effective peer-to-peer facility for distance and open learning involves tackling serious technical and social challenges.

In this paper we discuss the main characteristics of such an environment, *ModellingSpace* (MS) a distributed application facilitating peer-to-peer interaction, which comprises a suite of interconnected tools to support collaborative modeling activities of partners at a distance over low-bandwidth connections (Avouris et al. [3]). 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, which are the building blocks of models, building and exploring models that are constructed using the primitive entities, synchronous and asynchronous interaction of users, collocated or at a distance, who collaborate in building models. The open character of MS means that the users have access to an open set of primitive entities that can be used for building their models.

Mmodeling activities using MS have been originally conceived for primary and secondary education students, with special emphasis on facilitation of semi-quantitative reasoning over the constructed models (Dimitracopoulou et al. [7])., However a number of recent pilot studies have also indicated the effectiveness of use of such environment in higher and adult education settings and in particular in the context of open and distance learning (ODL) courses. It has been shown that MS can facilitate no-mediated peer-to-peer interaction of students (Avouris et al. [4], Komis et al. [11]). As a result, the MS prototype has recently attracted the interest of the open and distance learning community, since this environment can be used for concept mapping and graphical representation of complex models, which can be developed in various subject studies. Examples of such activities are the development of *Entity-relationship diagrams*, *flow charts* and other models in software

N.M. Avouris is with the Human-Computer Interaction Group of the Electrical and Computer Engineering Department, University of Patras, , GR-26500 Rio Patras, Greece (e-mail: N.Avouris@ee.upatras.gr).
M.Margaritis is with the Human-Computer Interaction Group of the Electrical and Computer Engineering Department, University of Patras, , GR-26500 Rio Patras, Greece (e-mail: Margaritis@ee.upatras.gr).
V.Komis is with the Early Childhood Education Department of the University of Patras, 26500 Rio Patras, Greece (e-mail: Komis@upatras.gr).

engineering, computer programming and database courses (Fidas et al. [9]), development of *Concept maps* of theoretical courses in various subjects (Margaritis et al. [15]). This kind of activities are planned to be introduced in some of the Computer Science courses of the Hellenic Open University as from the next academic year, in pilot use.

Some of the constraints related to such use of synchronous collaboration support tools are the cost of equipment and network bandwidth, which can be often limited for students of open and distance learning courses. Also other technical issues that need to be tackled are related with overcoming the restrictions imposed by Internet Service Providers (ISP) to peer-to-peer connections for security reasons. In the following section a discussion of the ModellingSpace architecture is included that addresses these issues.

II. MODELLINGSPACE ARCHITECTURE

The typical activity environment of ModellingSpace is shown in Fig. 1. On the left-hand side column of Fig. 1 a library of entities is shown, while on the right hand-side a library of available relations (links) is included; these are the building blocks for modelling. In the main window of fig.1 a simple model involving two entities and a link relation between them, defined through a graph, is shown.

It should be noticed that there are many kinds of entities in ModellingSpace. Abstract entities can be represented by textual descriptions, while others can have more dynamic behaviour. Some entities may even be represented on the work surface through multimedia files, e.g. images and video. Interconnection of such entities through the available links can result in complex models.

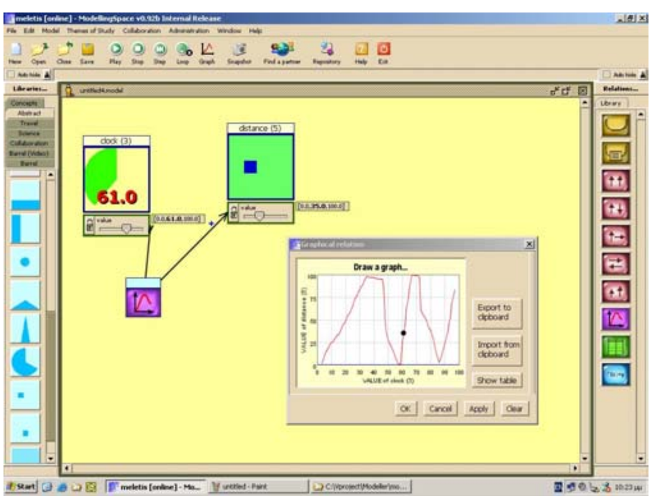


Fig. 1. The ModellingSpace environment

In collaborative model-building, the items included in the main window of Fig.1 need to be reproduced simultaneously in all collaborating partners windows. This is achieved through an object replicating architecture, which allows exchange of just control messages and maintains the content of the libraries in all partners' sites. This mechanism is discussed in more detail in the following sections.

In case that a complex entity is used by one of the collaborating partners and cannot be found in peers' entity libraries during modelling, a need arises to transmit this entity to collaborating peers in order to synchronize the peer applications. In this case the exchanged messages are not just control messages, but also larger files. This may result in relatively long download times. A solution for this problem that has been implemented in MS, is to send the light control messages directly to the peers (chat and change of state), including the structure of new primitive entities, while the multimedia files that may be associated to these entities, to be send through a server to the requesting peers, without creating disruption to the rest of the group. This is discussed in more detail in section III.

A. *The shared work surface design*

Synchronous collaborative problem-solving is often based on a shared work surface (Dix et al. [8]). As a result, communication among partners is done through the constructed artifact, found on this surface, e.g. a model under construction or the representation of a solution to a given problem. This is done in effect when one users' manipulation of the objects in this surface is observed by the peers. This indirect way of communication can be as important as direct communication (Avouris et al. [1]).

Various architectural decisions are related to the design of the shared work surface. One possibility is to apply a strict WYSIWIS (what you see is what I see) approach in the main work surface window. As a result the activity in this area is faithfully reproduced in all users' workstations. So most of communication and reasoning of users is based on this shared viewpoint, which becomes the main grounding mechanism of dialogue and through which eventually common understanding can occur. However additional operations outside this shared workspace may also be performed independently by partners involved, a model-level coupling approach according to Suthers [17]. This way a more relaxed coupling of partners is achieved.

Fig. 2 shows a typical collaborative activity, which involves two partners at a distance. These two partners interact through a reliable TCP connection, using the socket interface for client-to-client communication. A set of primitives have been developed, implementing the semantics of the protocols described in this paper. In contrary to some other collaboration applications, in which emphasis is in communication (argumentation tools, decision making etc.), in our case the distant partners collaborate mainly by sharing the model in the work surface, which thus becomes a cognitive space. In this case the communication through the artifact is important, where one participant's manipulation of shared objects can be observed by the other participants. A key requirement in this context is to support sharing of a view of the model in synchronous modeling activities over low bandwidth connections, as is often the case with open and distance learning courses students, who access the Internet through dial-up connections. In contrary to other shared workspace environments in which heavy graphical information is exchanged among partners, in MS we use a replication of the libraries of primitive entities and tools. As a result only light messages are exchanged among the partners. These are of the following three types:

(i) Change of state control messages, shown as (b) in fig.2. For example the following message concerns move of object *Sticky_note_2* to a new position on the screen. This is transmitted to the collaborating peers and the local client engines effect the move of the object.

```
<message>
       <ID>Move object</ID>
       <user>George</user>
       <objectID>Sticky_note_2</objectID>
              <attributes>
                     <x>100</x>
                     <y>250</y>
              </attributes>
</message>
```

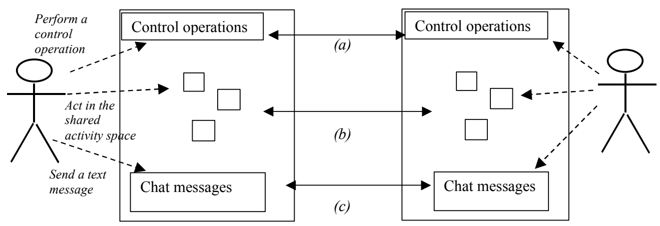


Figure 2. Peer-to-peer Collaborative environment. The exchanged information concerns (a) coordination control messages (b) shared workspace state-change control messages and (c) chat messages.

(ii) In addition support of direct communication among the participants through an instant text messaging tool (chat) is shown as (c) in fig.2. Text-based communication is a decision which has been preferred to audio or video communication, as it is more effective in low-bandwidth connections, while use of chat is only supplementary to the main communication channel, i.e. the observation of activity in the shared activity space. In the reported studies, it was found that the text-based communication was considered adequate by the participants and was proven effective in terms of the quality of the produced solutions, for the typical problem-solving activities of MS.

(iii) Finally control messages are exchanged which relate to coordination of the activity, like messages concerning locking of the activity space by one partner. These are shown as messages (a) in fig.2. A more thorough discussion of the alternative coordination mechanisms is included in section III.

The design of the MS environment has been a challenging process. In particular we have been concerned with mechanisms for coordinating the activity and with mechanisms for overcoming the problems imposed by firewalls and proxy servers, which make establishment of point-to-point connections difficult. This is a particularly important concern for students of open and distance learning courses whose access to the Internet is usually through dial-up connections to ISPs who impose many limitations to peer-to-peer connections. In the following we describe the main characteristics of the architecture of the system that has overcome these challenges.

III. Peer-to-peer interaction protocols

The MS architecture is based on a thick client component, which contains a number of interoperable tools. Even synchronous collaboration is effected through peer-to-peer interaction. However the proposed architecture contains also a server node (Community server), which is used as a common repository of information and as a central means for registration and authentication of users participating in collaborative interaction. This Community server should be maintained by the Distance Learning course supplier. Many issues related to security and asynchronous interaction can be solved through this server, as proposed by many collaboration support systems, e.g. see (Constantini et al. 2001). Additional functionality of the server involves support for asynchronous collaboration (asynchronous exchange of messages and files through a *tray mechanism*, logging of asynchronous interaction, etc.), tracking of physical address of online users, information on presence support, i.e. inform users on availability of their peers for synchronous interaction and support for smaller communities (the groups), where most of the activity takes place, by providing them with private space in the repository and private asynchronous interaction support. Finally, these Community Support Tools provide services like group management, session management, registration and login of users, etc, see also Avouris et al. [3].

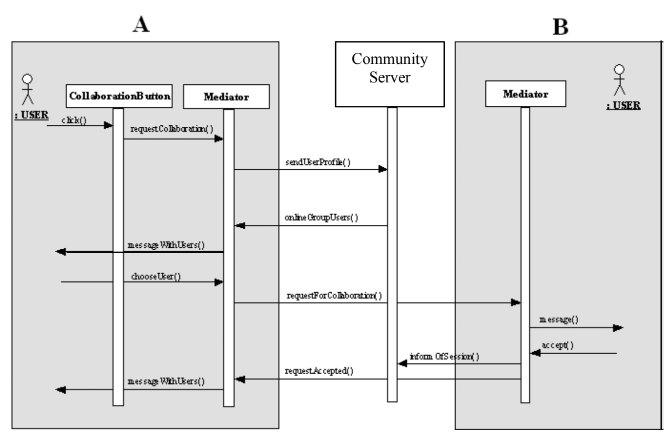


Fig. 3. Initiation of Collaboration session between peers A and B

The peer-to-peer collaborative session, is established as follows: The user activates a request for synchronous collaboration, selecting an individual user or a group of users from the on-line users in the server users space, as shown in the interaction diagram of fig.3. The system checks if a model is in the process of creation in the activity space, in such case the system informs the users that the activity space should be cleared before collaboration can be initiated. The system sends the request to the user(s).

The reply of the user(s) is either acceptance of the request or rejection of the request, if no reply is provided within a time limit a "reject collaboration" is assigned to the particular user. If the collaboration request is done in the frame of an existing group, then a collaboration session is established (logging parameters, continuation of previously suspended collaboration session). In fig. 4 the dialogue window of selection of collaborating partners is shown. In this figure, three users make up group *Review3*. At the instance shown in the figure, two of these users are offline and one is online.

If the request is accepted by some of the users, the collaboration panel is activated, as shown in fig.5, and a chat window is created. The collaboration panel permits the following operations: *Request the key, pass the key, send a model*, disconnect, while it provides a *collaboration awareness* mechanism.

If the collaboration session is generated by a group coordinator, the coordinator can decide on the collaboration protocol (round robin, key passing mechanism, role playing protocol, see also discussion on coordination mechanisms in section IV). Once a collaboration session is initiated no more users can join in, while the coordinator can join in and leave the group at any time. Collaborating users can quit the session at any time. This is acknowledged to the other partners. While the role of a coordinator has been foreseen in this scenario, it is usually the case in Open and Distance learning courses that the peer-to-peer interaction is not mediated by a supervisor, in this case one of the peers takes the initiative to start a collaboration session.

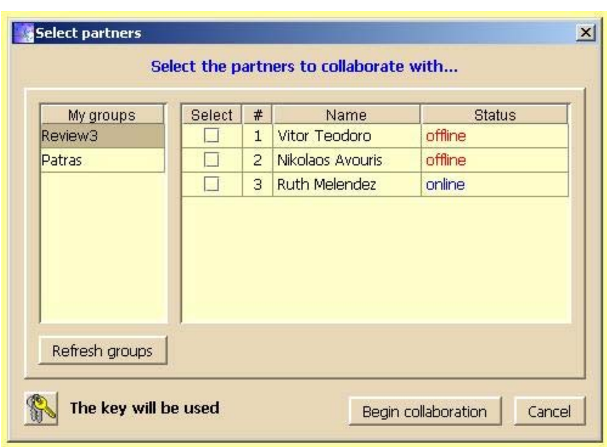


Fig. 4. Selection of partners for synchronous collaboration using the Community server (Group Review3 has been selected).

A. *Exchange of objects during collaboration*

As described in the previous section, the Community Server plays a role only during initiation of collaboration. Peer workstations' synchronization is achieved without intervention of the server. The mechanism is based on a set of reactive agents, which try to achieve synchronization with the corresponding agents of the peer host, based on a stimulus–response model. So in a joint problem solving activity each object and each relation introduced, act as reactive agents. The behavior of each agent depends on whether it is on the active user's side or on the passive user's side at a specific point in time. If it is on the active user's side it monitors user events that are related to the particular object (movement, changing of properties, deleting etc.), and sends these events to the equivalent agent on the passive user's side. This is achieved through the *Mediators*. The size of the event-passing messages is variable and depends on the kind of actions of the active user. However in most cases it remains very small, permitting good run time performance. When the Mediator of the passive user's side receives the message, it decodes it and informs the equivalent agent who acts accordingly.

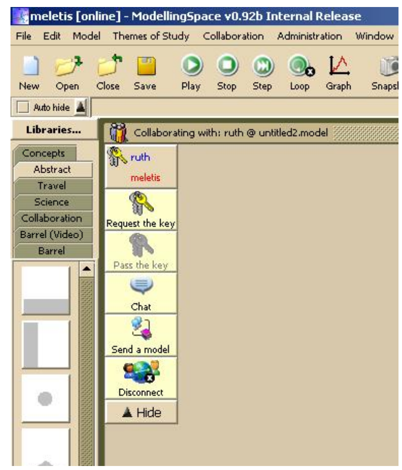
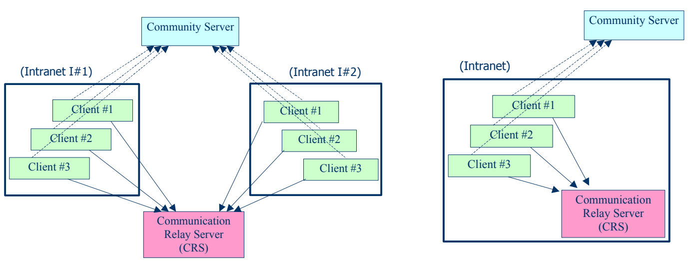


Fig. 5. Collaboration panel

This necessitates that the objects present in the Activity Spaces of two collaborating partners are identical. However, as discussed earlier, there is a possibility that two users are in possession of different primitive library objects, due to the open architecture of the environment. So there can be a case when the active user A adds an object into the shared activity space, which does not exist in the library of user B. In this case it is necessary to update the library of user B at run time with the missing object before proceeding any further. This is done transparently from the users as follows: When user A inserts the new object O_i in the Activity Space, Mediator A informs Mediator B about the addition of the new object, sending the appropriate message with the object's unique ID. Mediator B searches the local Entity Library for O_i If this object does not exist on host B then Mediator B asks A to send a copy of object O_i before proceeding any further. Mediator A sends the object, and waits. During this activity the user actions in the shared Activity Space are suspended and a message is displayed that the peer library is updated. After the sending is complete Mediator B informs Mediator A that it has received the object and the activity can proceed. The object multimedia attachments can be send either directly as shown in figure 4 or through the server if the size of the multimedia files are too large and can disrupt activity for both partners for too long. In the latter case the message is send to the Community Server with the ID of the object, the server sends the object to the user. If the object does not exist in the server, it is downloaded, transparent to the two users from the library of user A.

B. Communication through proxy servers

One of common problems in p2p protocols is that of overcoming the restrictions imposed by firewalls and proxy servers who do not allow point-to-point connections to not-trusted sites, while dynamic allocation of IP addresses creates difficulties in establishing reliable connection across ISP boarders. A solution to this problem is the definition of a trusted Communication Relay Server (CRS), residing in a host with public IP address. The role of this server is to relay the exchanged control messages to collaborating partners. This component of the MS architecture has been used effectively overcoming the above problems, permitting control message tunneling, traffic coordination, improving client security, since the communication is done only towards the trusted CRS node.



(a) collaboration across intranets (b) collaboration within intranet

Fig. 6. Use of Communication Relay Server

While the introduction of the CRS component solves these problems, in effect this solution lead to an implementation of the p2p protocol through a client server mode which defeats some of the advantages of the p2p approach. For instance the existence of a central CRS server creates a bottleneck in communication and does not scale up. A more flexible approach to this problem, that has been lately used in MS, has been to let the final user decide on the CRS to use for collaboration. In effect we have included in every installation of MS a copy of the CRS, so any host running MS software can become a relay server. A default relay server resides in the Project Community server (www.modellingspace.net), however if a user decides to start a collaboration session using his/her own host as relay server, this can be done by setting up the appropriate parameter in the MS environment. This can be the case that the collaborating partners are located in a local area network, so that is more effective to communicate using one of the local hosts as a relay server, as shown in case (b) of fig.6.

Finally the possibility of overcoming completely the Community Server and use just a local Communication Relay Server for synchronous collaboration is also allowed by this flexible architecture. This is the case of a group of users in a local area network with no connection to the outside world, who wish to collaborate using the p2p protocol. In the latter case, however some of the services of the Community Server are missing, i.e., the history of group collaboration cannot be retrieved, while presence info about group members is not available.

In a typical Open and Distance Learning setting, a number of CRSs should be provided to users in specific geographical areas in which large numbers of ODL students reside.

IV. Coordination of Collaborative Activity

The coordination of partners' activity in the shared activity space is a very important aspect of the architecture. In general, the coordination mechanism of the activity in the shared workspace can take many forms. See Dix et al. [8] for a survey and a discussion of alternative approaches. Some of these approaches impose no particular control, i.e. any member has his/her own pointing device and can manipulate objects in the activity space or write on the whiteboard. This is claimed that may result in chaos with participants ending up in writing one on top of the other and canceling each other's actions. Other approaches propose floor control mechanisms, involving the existence of a coordinator, various floor control protocols, like round-robin etc, or protocols of explicit request/ concession of the floor with time constraints. For instance inactivity of the floor owner for more than a certain time can release the floor.

Two alternatives have been provided in relation to coordination mechanisms for ModellingSpace design. The first mechanism involves a token, the *Action Enabling Key*, which is owned by one of the participants at any given time. This key owner imposes a lock on the shared activity space. The owner of this token can act in the shared workspace, while the other participants just observe this activity. This mechanism is supported by key request, key pass, key accept, key reject functions, shown in the panel of fig.4. The interchanged coordination control messages are shown as connection (a) in fig.2. The effectiveness of this approach has been studied in various experiments, see Fidas et al. [9] and Komis et al. [11].

An alternative that has been also implemented, proposed especially for small groups of partners, involves lack of such floor control mechanism. The partners can manipulate parts of the model at any time during problem solving. For reasons related to distributed data consistency, only a temporary locking mechanism of objects selected by one partner is imposed during an operation, as shown in fig.7(b). The coordination of activities is left to the partners to decide in this case. So, the activity of a partner cannot be inhibited and no conflicts can occur over key possession. Nevertheless, in this case, implicit social protocols of organization need to be established by the users, as discussed in Avouris et al. [2], in order to facilitate coordinated group activity, since explicit coordination is not imposed.

Early experiments with this kind of floor control mechanism, have indicated that it may improve reasoning about action, as partners need to reason and negotiate during key requests. In the experiment reported by Avouris et al. [4] the effect of this mechanism on problem solving was studied, by comparing the performance of two groups of users one of which used this mechanism while the other used no explicit floor control. A side effect of the no-floor control case is observed when two users attempt at the same time to handle the same object. In this case the final state of the specific object dependents on the order of release of the lock on the object by the partners involved, as shown in fig.7(a).

A. Direct communication and sticky notes

In this section the additional communication mechanisms provided by MS are discussed.

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

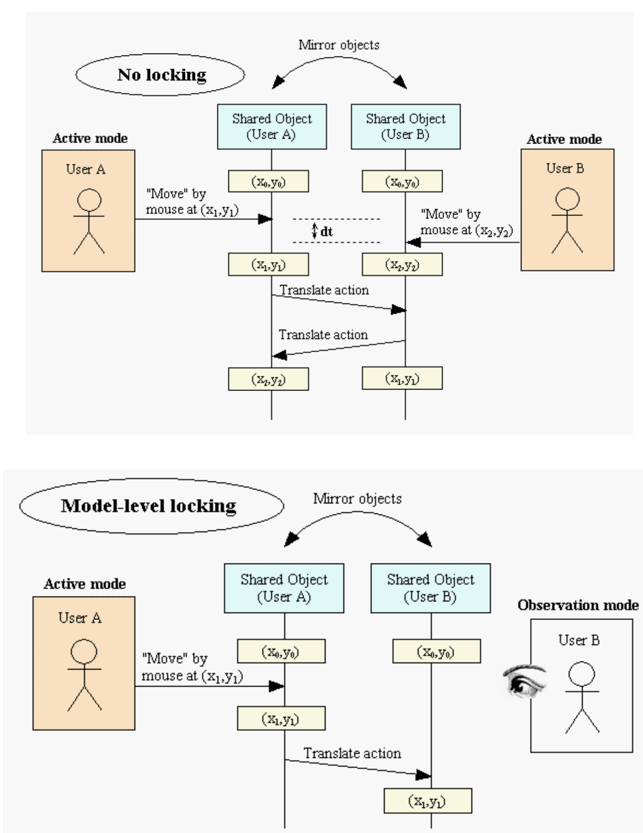


Fig. 7. Locking of objects in the shared space.

This chat tool, which is activated from the collaboration panel, is equipped with dialogue openers, i.e. phrases like "I agree with…", "I object to…", " I think that…", which can be used to open a chat message. This way the user can select the opening phrase of the message and thus classify indirectly the speech act. There has been 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, Soller [16]

Other means for exchange of text messages are the *sticky notes* (text containers positioned in the work space). These 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 work surface, can be related to an object in this space and through this a comment by one of the partners can have a permanent effect.

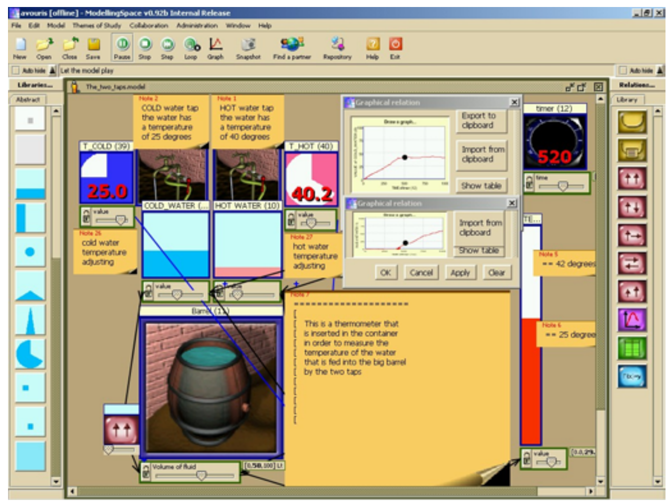


Fig. 8. Example of a model with use of various sticky notes

In the model of fig.8[1], the sticky notes are used for indicating the various primitives used, as well as to mark the critical temperatures on the thermometer on the right-hand side of the model. The permanent effect of these items on the model is an aspect that has been investigated and discussed by Fidas et al. [9].

V. Conclusions

In this paper we discussed a peer-to-peer architecture that permits real-time collaborative modelling at a distance. This architecture is particularly suitable for synchronous collaboration of students of open and distance learning courses. This is because the proposed architecture can be implemented in low-bandwidth networks, it does not necessitate any special equipment, like in the case of video-based teleconferencing systems, while it allows exchange of information between peers who are behind firewalls. In particular, the proposed architecture involves exchange of just control messages for maintenance of effective

[1] The model of Fig.8 simulates the effect of two taps filling with water of different temperatures a container.

WYSIWIS (what you see is what I see) of the shared workspace, as well as text chat messages for direct communication and coordination control messages. These messages are at the most a few bytes long (4 to 100 bytes) and therefore can be exchanged without disruption of interaction even under low bandwidth peer-to-peer connections. The effectiveness of this approach has been proven through a number of case studies in authentic collaborative problem solving settings, reported by Avouris et al. [4], Komis et al. [11], Margaritis et al. [15], in which alternative cooperation schemes have been implemented.

The proposed architecture is characterized by a great degree of flexibility, as it permits use of various coordination schemes and levels of locking of objects in the shared activity space, while it proposes a flexible scheme of communication relay in order to overcome security problems and restrictions imposed by ISPs.

An additional advantage of the proposed approach is the possibility of monitoring and analysis of collaboration logging data by instructors. A number of tools and various methodological frameworks have been proposed for this purpose (e.g. Avouris et al. [2]). This possibility can be further facilitated by automatic analysis tools and techniques that can be applied on large amounts of data that can be collected, see Xenos [18].

The solutions discussed in this paper are applicable in a wide range of synchronous collaborative scenarios and the presented environment is particularly suitable to be integrated in the toolbox of open and distance learning courses and organizations like the Open University and other ODL service providers. However, the effectiveness of large-scale implementation of this approach and its impact on the community of students of ODL courses is still under investigation.

Acknowledgment

The reported work has been partly funded by the IST-School of Tomorrow Project IST-2000-25385 "Modelling Space". In this project participate 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, Vol. 19, 2, pp. 147-167.

[2] Avouris N., Fiotakis G., Tselios N., Komis V., Tools for task-based interaction and collaboration analysis, Proc. HCI International 2003, Crete, June 2003.

[3] Avouris N., M. Margaritis, V. Komis, A. Saez, R. Melendez, (2003) ModellingSpace: Interaction Design and Architecture of a collaborative modelling environment, Proc. of 6th Conf. Computer Based Learning in Science (CBLIS), 2003, Nicosia, Cyprus.

[4] Avouris N., Margaritis M., Komis V., (2003b), Real-Time Collaborative Problem Solving: a Study on Alternative Coordination Mechanisms, IEEE ICALT 2003, Athens July 2003.

[5] Bouras C., Gkamas A. Tsiatsos T. (1999), Distance Learning environment using advanced services over the Internet, 3rd IASTED Conf. Internate and Multimedia Systems and Applications, Nassau, Bahamas, October 1999.

[6] Constantini F., Toinard C., (2001), Collaborative Learning with the Distributed Building Site Metaphore, IEEE Multimedia, July-Sept. 2001, pp. 21-29.

[7] Dimitracopoulou A., Komis V. (2003). Design Principles for an Open and Wide MODELLINGSPACE for Learning, Modelling & Collaboration in Sciences and Mathematics, In (Ed) C. Constantinou "Computer Based Learning in Sciences", Proceedings of Sixth International Conference CBLIS, 5-10 July, 2003, Nicosia, Cyprus.

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

[9] 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.

[10] Kato H., Yamazaki K., Suzuki H., Kuzuoka H., Miki H., Yamazaki A., (2002), Designing a video-mediated collaboration system based on a body metaphor, CSCL 2 Carrying forward the conversation, Koschmann T. et al. (ed.), pp. 409-424, LEA, Mahwah, NJ.

[11] 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.

[12] Komis V., Avouris N., Fidas C., A study on heterogeneity during real-time collaborative problem solving, In (Ed) U. Hoppe, Computer Support for Collaborative Learning: Designing for Change in Networked Learning Environments, CSCL 2003 congress: 14-18 June 2003, Bergen, Norway.

[13] Lethin R., (2003), Technical and Social Components of Peer to Peer Computing, Communications of ACM, Special issue on P2P Computing, vol 46, 2, pp. 30-32, February 2003.

[14] Lopez P., A. Skarmeta, (2003) ANTS Framework for cooperative work environments, IEEE Computer, vol. 36,3, pp. 56-62, March 2003.

[15] Margaritis M., N. Avouris, V. Komis, A. Dimitracopoulou, (2003) Real-time collaborative modelling over low-bandwidth networks, Proc. of CSCL2003, Bergen, 2003.

[16] Soller, A.L. (2001) Supporting Social Interaction in an Intelligent Collaborative Learning System. Int. Journal of Artificial Intelligence in Education, 12(1), 40-62.

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

[18] Xenos M., (2003), Modelling of the behavior of computer science students with Bayesian networks, in Proc. 2nd Conf. On Open and Distance Learning, A. Lionarakis (ed.), pp. 318-329, Open University Press, Patras.(in Greek)