Design of collaboration-support tools for group problem solving

C. Fidas[1], V. Komis[2], N.M. Avouris[1]
University of Patras
[1]Electrical and Computer Engineering Department - HCI Group
[2]Early Childhood Education Department
GR 26500 Rio Patras, Greece
Fidas@ee.upatras.gr, Komis@upatras.gr, N.Avouris@ee.upatras.gr

SUMMARY
This paper discusses human-computer interaction in the context of collaboration-support environments. In particular our experience with the design and experimentation with tools to support collaborative problem solving is described. Alternative design options are studied and their effect on synchronous collaboration at a distance is discussed through observation of users' performance. The use of collaboration analysis tools, which have been developed in the frame of this research are also discussed.

KEYWORDS: Collaborative problem solving, human-computer interaction, synchronous collaboration, computer-supported collaborative learning.

INTRODUCTION

This paper focuses on our experience with design and experimental use of tools to support synchronous collaborative problem solving. The design of such tools is not a trivial process. The relevance of this activity has increased during the last years due to the increasing maturity of the enabling technologies which make collaborative problem solving feasible, the increasing number of collaboration support applications that have created a corpus of design knowledge and expertise and the social pressure for further development and use of this technology in many areas of human activities like engineering design, training and education. The Design of Collaboration support environments is a complex process. The shifting technological background and the limitations of the narrow bandwidth of existing technology make design of tools particularly difficult process.

Special attention is provided here in the design of collaboration support tools, in the context of a collaborative problem solving environment, where collaborating peers build a solution to a problem by manipulating objects in a common workspace and interact through a chat tool. In particular four alterative interaction designs concerning these tools are studied and experimentally evaluated in this paper. These relate to (a) the mechanism developed to inform the collaborating peer on the current status in the **chat tool**, (b) the usage of the **sticky notes** mechanism permitting interleaving of text messages and objects in the working space and (c) the study of the impact of the **objects locking mechanism** applied in the common working space.

In the following section of the paper an overview of existing collaborative problem solving architectures is provided. In section 3 the R2 environment used in the frame of the current research is briefly described. The design of collaboration-support tools of R2 and their evaluation are discussed in section 4. Special emphasis is provided in the development of an analysis framework for collaborative problem solving, which has supported the evaluation of the developed tools. Some reference to these tools is made in the final part of the paper.

GROUP PROBLEM-SOLVING ENVIRONMENTS

A number of tools have been developed and used during the last years that support synchronous collaborative problem solving and interaction. While some tools (e.g. Netmeeting, ICQ etc) support general-purpose collaboration and interaction, a number of tools have been developed for collaborative learning and problem solving. Some typical examples are CSILE (and its latest version Knowledge Forum®), Belvedere, CoVis, DIVE etc. The key characteristics of these environments are discussed in this section.
CoVis provides students with a number of collaboration and communication tools (desktop video teleconferencing, shared software environments for remote, real-time communication, a multimedia scientist's notebook and scientific visualization software) so that they can develop dexterities similar to those of scientists.
Knowledge Forum® contains tools for the development of shared databases by groups of individuals creating communities with similar interests. Furthermore it provides communication tools, development of concept networks and offers tools for viewing the knowledge base from multiple perspectives.
Belvedere constitutes a representational tool for the acquisition of collaborative dexterities in the investigation of real scientific problems. It belongs to the category of those learning environments that mediate collaborative

learning interaction and communication of the externalised knowledge through an appropriate tool (Suthers, 1999).

The *Distributed Interactive Virtual Environment DIVE* (Hagsand, 1996) enables several users in distant locations to share a virtual space. It can be used to perform virtual experiments and carry out creative tasks. This is an experimental platform for the development of virtual environments, user interfaces and applications based on shared 3D synthetic environments. DIVE is especially tuned to multi-user applications, where several networked participants interact at a distance. It is based on a peer-to-peer approach with no centralized server, where peers communicate by reliable and non-reliable multicast, based on IP multicast. Conceptually, the shared state can be seen as a memory shared over a network where a set of processes interact by making concurrent accesses to the memory.

The framework system proposed by Muehlenbrock et al (1997) is characterized by the combination of intelligent support with interactive learning environments, by the provision of reusable components and by a distributed multi- agent architecture.

In addition, during the last years many software tools that support construction of diagrammatic representations have been developed. Most of them however do not allow collaboration between users through their computational environment and only few allow communication in the form of file exchange in a synchronous or an asynchronous way. Examples of the latter case are Representation Tool 1.0, Belvedere, MindManager®, Inspiration® 6.0.

So support for synchronous collaboration of students with the aim of constructing diagrammatic conceptual representations or other shared constructions into a common space is a new challenge to tools of this nature. Based to this perspective, Representation 2.0, (Komis et al. 2001), an innovative environment supporting collaborative creation of diagrammatic conceptual representations has been developed. This environment has been used experimentally to support collaborative problem solving under real educational conditions. This environment is presented in the following.

THE *REPRESENTATION* 2 (R2) ENVIRONMENT

Representation- version 2.0 (R2) is an educational software environment supporting collaborative learning. R2 has been developed recently as an evolution of Representation (v 1.0) in the frame of a European research project. R2 has been used experimentally both in collaborative and single-user mode for supporting building semantic representations in various educational contexts and for experimentation and study of collaborative learning. Typical user view of the R2 environment is shown in figure 1.

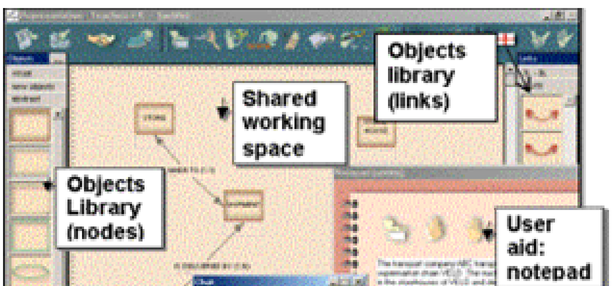


Figure 1. The R2 Environment

The R2 environment provides tools for individual and collaborative expression of knowledge through diagrammatic representations of abstract concepts and relations that link them. The primary objects supported in the diagrammatic representations built with R2 are thus node objects (concepts) and link-objects that connect the nodes. Libraries of such objects are already provided to the users of R2. These libraries can be extended by the users. The tool has been used for expressing in a diagrammatic way abstract representations like semantic diagrams that play an important role in collaborative problem solving (Swartz 1995). So R2 is a typical environment that can be used in learning through collaborative problem solving based on these diagrams.

The tool provides facilities for synchronous interaction between collaborating partners engaged in problem solving. It supports simultaneous development of diagrammatic representations of collaborating partners through the use of a shared Activity Space and through the provided dialogue and negotiation tools discussed bellow.

R2 permits expression and investigation of ideas and understanding of students through the manipulation of simultaneous multiple representations (Suthers, 1999) of analogical and symbolic form. The environment provides users with the capability to manipulate and link the images and symbols, which represent concepts. Users manipulate directly the elements of the diagrams that belong in already provided libraries or new ones that can be added as images and symbols, strengthening this way the openness of the software. The R2 environment allows and favours the communication and exchange of diagrams while at the same time supports their joint creation and manipulation. In this context, learning interactions (Suthers, 1999) between students and representations, between peer students, between students and teachers or researchers are favoured.

In figure 1, the collaboration space is shown in the central part of the picture. In the two sides, the libraries of objects and libraries of links are shown. In the same figure the communication tool (chat) and the handler through which control of the Activity space can be affected, are also shown.

The primary objects that the R2 user can handle for building semantic diagrams, are objects and links. The objects can either be analogical representations of known objects (images of real life objects) or abstract objects in which the user can insert textual or iconic representation of concepts to be represented. In figure 1, on the left column some available library objects representing computer components are shown, while in the activity space abstract objects have been inserted. The links can be either typed (named) or untyped. Typed links indicate better understanding of the model by the diagram developer. (Fisher, 1990). The diagrams developed through R2 can be made of multiple levels: It is possible to associate a new diagram of a lower level to an object. This multi-level representation is easily navigable by the user through clicking on object images. The multi-level diagrams created through this tool can be complex conceptual constructs.

A log file of the diagram creation process is automatically created and saved together with the diagram. This can be used by the teacher /researcher as a cognitive tool providing useful information regarding the development of the student involved. These log files have been the main source of information for our study, as discussed in the following sections of the paper.

Two different types of users of R2 are identified. These are the students and teachers/researchers. The latter have access to the functionalities of the teachers/researchers support component, the analysis tools, etc. R2 distinguishes these two user groups during login.

Communication support tools.

R2 can be used either as a stand-alone educational environment, or as a collaboration support tool. As a collaboration support tool, R2 can be used in both asynchronous and synchronous way. The chat room facility permits exchange of free-text communication messages between collaborating partners. Also a diagram exchange tool has been implemented. This is both a synchronous and asynchronous communication tool. If the recipient of the diagram is on-line during transmission, the diagram is sent directly to the receiving partner. If the recipient of the diagram is off-line, the diagram is stored in an ftp-server and when the recipient is connected, the transmission is completed.

The shared Activity Space is a drawing space of synchronous collaboration, in which one of the two collaborating partners can insert and modify primary objects (concepts and links), organize multi-layer diagrams etc., through direct manipulation. The supported protocol of interaction is described here: When connection between two partners is established, following a "request for collaboration" of one partner, accepted by the other, a copy of the action space is build and maintained in both parts involved until the connection is terminated by one of the two partners. The two partners can exchange roles, playing either the passive or the active role. The active partner is the one who can manipulate objects in the action space. These actions generate messages transmitted to the passive partner, thus reproducing the same effect at the screen of both workstations. So R2 supports a shared WYSIWIS (what you see is what I see) environment. A mechanism is established for exchange of roles. The metaphor used is that of "passing the key". The holder of the "action-enabling key" is the active partner. Through this key request/ key accept/ key reject protocol the active role can change at any point during collaboration, provided that the passive partner requests the key and the active partner accepts the request. An implication of this "key exchange" protocol is that deadlocks can be created in cases when one partner cannot proceed with problem solving and at the same time refuses to pass the key over to the other partner. Such situations did occur during the reported experiment. Despite this, the protocol maintains clear semantics of actions and roles in the shared activity space and therefore is considered essential part of the architecture. This conclusion seems to be in agreement with the view expressed by researchers of similar environments, see (Soller, 2001). A variation on the ownership of the solution object has been subject of research as discussed in the following section.

DESIGN OF COLLABORATION TOOLS

The original design of the collaboration tools described in the previous section has been tested with the participation of real users. As a result of the original experiment a number of modifications and improvements to the collaboration protocols have been made. These have been subsequently tested in an experiment against the original design.

The limitations of the chat tool

Text-based synchronous conversation (chat) tools have been used for a number of years as a means of synchronous interaction. The tools in their current form are still quite primitive, despite the widespread use in collaborative environments and many attempts to develop graphic chat tools, cartoon-based chat, character animation based tools etc (e.g. Kurlander et al, 1996 etc.). The current chat tools fail to convey many basic social cues (Viegas, Donath, 1999). The limitations of the original tools are due to the limited expressiveness of text to convey back channel information (Dix et al. 1998, section 14.4) the difficulty on establishing grounding mechanisms, the lack of turn-taking and sequencing mechanisms. Lack of deictic references etc. Additionally, in the case of users of young ages, who have limited development of their language expressiveness and typing skills these problems seem to become more critical.

A number of modifications to the original chat tool have been attempted during the reported experimentation. These relate to (a) the provision of back channel information on current activities of the collaborating peers. and (b) the possibility of interleaving text messages with actions in the shared working space. It has been observed during dialogue analysis that through these modifications the expressive power of the tools has been improved and the ambiguity in the dialogues has been reduced.

A short description of the affected modifications is described here.

(a) *Feedback on peer activity*

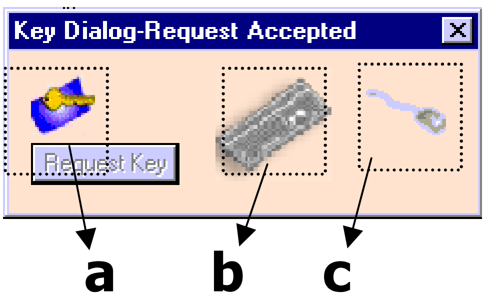


Figure 2. *The collaboration control panel: (a) The action key request/granting tool, (b) the feedback on peer keyboard activity and (c) the mouse activity.*

In the collaboration control panel of R2 two new images were added. These, shown as (b) and (c) in figure 2, give a feedback to the user on the activity of the collaborating peer. The keyboard image is blared when no keyboard activity occurs at the other end, while it becomes more intensive when the partner is typing in the chat tool. Image (c) has similar behaviour in relation to the pointing device, e.g. it is highlighted when dragging and moving of objects occurs by the collaborating partner. This mechanism has reduced the ambiguity observed in turn taking and the interleaving of multiple dialogue threads, often observed in text-based dialogue tools.

(b) Deictic text communication

A second modification was related to the introduction of the possibility to introduce deixis in the text-based communication. In earlier versions of the environment, the lack of deictic reference was obvious. The users were often referred on objects of the common working space in the chat tool. Dialogues like the following occurred often.

- Move the blue object higher
- Which blue object the one on the right or on the left?
- No I do not mean the right one, I mean the one bellow…

An alternative introduced in the current version of the environment was to insert text messages in the working activity space in the form of sticky notes.

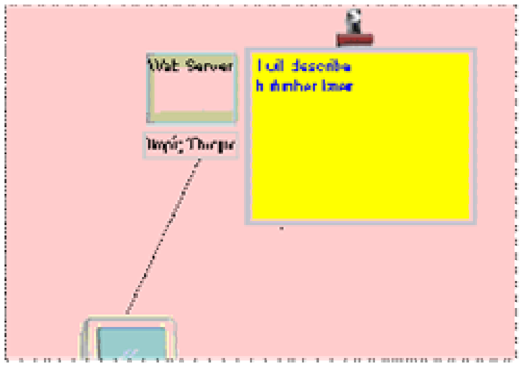


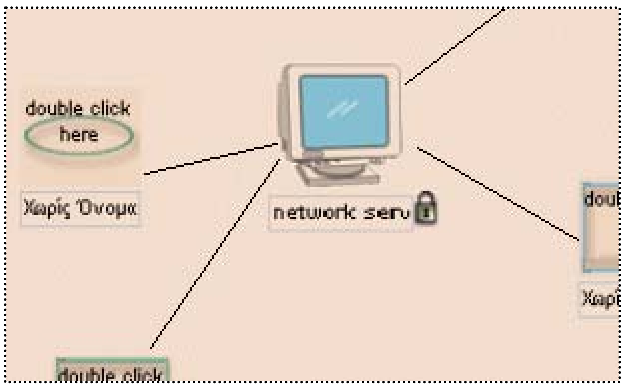
Figure 3.*Sticky note tool*

In figure 3 the use of the tool is demonstrated. One user has put a message in the form of a sticky note next to an object of the common working space, commenting the particular object. The deictic reference of this particular text is obvious through the use of this tool.

The ownership of collaborative solution

A second area of observed problems in collaborative problem solving environments is related to the ownership of parts of the common problem solving solution. The solution to the problem is built by introducing objects in the common working space by all partners involved. It has been observed however that often conflicts can occur during problem solving. In such case if partner A has introduced objects O1, O2 and partner B would like to modify them or delete them, a question arises on whether to grant such right to partner B. In the previous version of R2 the partners where free to modify any object found in the common working space, once in possession of the action enabling key. However through this protocol any partner can destroy the parts of the solution developed by their peers.

Figure 4. *Locking of objects in the common activity space*



An alternative mechanism has been introduced in this current version of the environment. An object locking mechanism has been devised. According to this the user when introducing a new objects can lock it, not permitting thus any modification by other users. For instance in figure 4 the object *Network server* has been locked by its owner. If another user would like to modify one of the properties of this object or to delete it, there are the following alternative options:

(a) A request can be issued to the object owner to unlock this object in order to be modified. This necessitates passing the key to the owner and back to the requesting partner
(b) The proposing the modification partner can suggest the modifications to the owner (through the chat tool or a sticky note) and ask the owner to affect them. However this option requires a long text dialogue, which can create disambiguities.

In the following section the experimental use of the developed tools is described and some of the findings are discussed.

ANALYSIS OF TOOLS USAGE

The Context of the Experiment

Two different versions of the tool were used by a group of 16 final year students of the ECE Department of the University of Patras. The task that the students had to do was to construct in a collaborative way a joint *concept map* concerning the Internet using the R2 tool. Eight (8) groups of two were built, four of which used version A (ver.A) and the other four used version B (ver.B) of the tool. Version B included the modifications discussed in the previous section. The experiment took place in the frame of the course concerning Advanced Software Development for the Internet. The students were encouraged to use any concept they though appropriate relating to the structure and operation of the Internet.

Results of the study

The main findings of the study are the following:
The version B of the chat tool was more effective in supporting conversation. Less ambiguity in dialogue and easier serialisation of dialogue was achieved through these new features of the environment. An extract of dialogue of GroupA2 is shown below.

37 : 45	1	Chat	Is it better now or before ?
38 : 08	2	Chat	I do not understand the result
38 : 27	2	Chat	Do you think we should leave it as it was?
38 : 46	1	Chat	It is more clear that the communication is two-way
38 : 56	2	Chat	Yes but we have to be consistent
39 : 09	1	Chat	What do you mean?
39 : 50	2	Chat	No it is not necessary to have two-way communication, we should put everywhere the same relation and get on
40 : 07	1	Chat	ok

In this extract interleaving of alternative threads of dialogue is observed, making interaction particularly difficult. This was not observed in the users of version B.

The solution ownership control mechanism was effective in inhibiting modifications of objects created by an actor. So in one instance of group A, when a partner started deleting objects in the common space, the owner replied with an angry text message:

```
WHY ARE YOU DELETING THEM THEY NEED OT BE
THERE!! (in capitals for emphasis)
```

However it should be observed that the dialogues of group B were longer, since one partner requesting a modification of somebody else's part of the solution needed to negotiate the modification first.
In the extract included in the following (group B4) partner 1 tries to attempt to convince partner 2 to modify part of the solution.

55 : 35			
56 : 27	2	*Chat*	You are right, but one does not need to link the two Bridges in order to affect the connection?
57 : 47	2	*Chat*	OK I need the key to change them
57 : 51	2	*Request Key*	

As one can observe from this extract partner 1, in possession of the key, needs to change a part of the solution that is owned by partner 2. In order to affect the modification, the partner needs to persuade p1. The dialogue is semantically rich, however the chat tool is perhaps not the most appropriate means for such dialogues, resulting in frustration of the users.

Tools for dialogue analysis

The analysis of the interactions produced in computer mediated collaborative learning systems provides a lot of useful feedback to researchers, in order to understand the collaboration process and to the collaborators themselves as a metacognitive tool. Recently much work has been done in the direction of developing tools to support this analysis process.
In our experiment the collaboration process has been analysed on the basis of the "Object- Oriented Collaboration Analysis Framework" (OCAF) according to which the objects of the solution become the centre of attention and are studied as entities that carry out their own history (N.Avouris, et al.2001).
Following the above approach many items of information, related to quantitative and qualitative analysis, enable us to understand issues such as the contribution of group members to the developed solution, the degree of participation of group members as well as the density of interaction.
The OCAF concept has been implemented by storing the interactions of the collaboration process in a database.
The entities involved are the collaborating users, the type of interaction and the objects making of the developed solution. Having the interaction stored in a database one can obtain various views of the process through appropriate queries.
An example of a specific group's analysis is shown in figure 5.

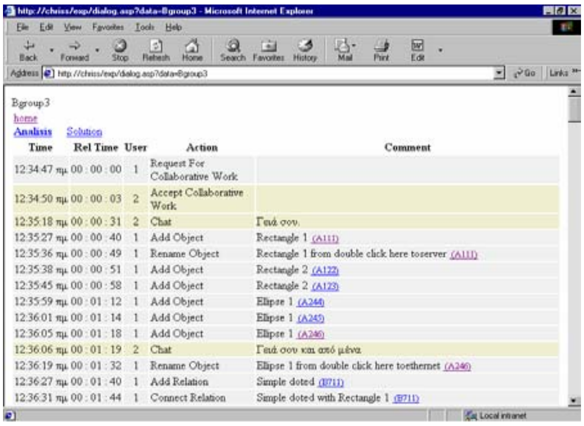


Figure 5. *A view of the interaction analysis (OCAF) tool*

In this figure one can see that the collaborator's interactions are classified according to time. Each object, which has been added to the activity space, has a link that allows us to see its history and additional information, like whether it is a part of the solution. Alterative views are based on actors, objects or the structure of the solution.

CONCLUSIONS

In this research the effect of alternative designs of the collaboration support tools in group problem solving was studied.
Following a qualitative analysis of the interaction and the views of the users involved as expressed in the questionnaires, the conclusion was that the feedback mechanism and the deictic text tool (sticky notes) resulted in more structured dialogues and less ambiguity in interaction. Less often the dialogues of groups B involved multiple threads and the turn taking mechanism was considered more effective. Characteristically, the users of Group A suggested in their comments as an improvement a mechanism that permits to receive a feedback on the actions of their partner.

As far as the communication protocol is concerned, the locking of the objects by their owner was found useful by the users involved because their ideas were safe but they found felt that at the same time this locking mechanism inhibited the flow of the problem's solution as every time one partner needed to change another partner's object there was need to get engaged in a long conversation on the rationale of the proposed modification. While this seems to have as a collateral effect deeper collaboration, on the other hand it reduced the direct manipulation style of interaction in favour of a more a dialogue interaction style The users view on this aspect was that there should be safety layers so that the users would be able to give the appropriate rights to their partners when mutual understanding was developed.

ACKNOWLEDGEMENTS

Special thanks are due to the students of the final year of the ECE Department who participated in the described experiment. Financial support from the Représentation Project, Educational Multimedia - Task Force (project contract 1045), of the European Union is also acknowledged.

BIBLIOGRAPHY

Avouris N., Dimitracopoulou A., Komis V. (submitted 2001). On analysis of collaborative problem solving: An object – oriented approach. *Int. Journal of Interactive Learning Research*.

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

Hagsand O. (1996). Interactive MultiUser Ves in the DIVE System, *IEEE MultiMedia*, Vol.3, No.1.

Kurlander D., Skelly T., Salesin D. (1996). *Comic Chat*, Proc. ACM SIGGRAPH.

Muehlenbrock, M.,Tewissen, F., & Hoope, H. U. (1998). A framework system for intelligent support in open distributed learning environments, *International Journal of Artificial Intelligence in Education*, 9, 256-274.

Soller, A.L. (2001). Supporting Social Interaction in an Intelligent Collaborative Learning System. *Int. Journal of Artificial Intelligence in Education*, 12, (in press).

Suthers D., (1999). Representational Support for Collaborative Inquiry. In *Proceedings of International Conference on System Sciences*, IEEE, Hawaii.

Komis V., Avouris N., Fidas C. (submitted 2001). Computer Supported collaborative problem solving: Interaction through diagrammatic and verbal communication. *Computers and Education*.

Schwartz D. L. (1995). The emergence of abstract representation sin dyad problem solving. *The Journal of Learning Sciences*, 4 (3), 321-354.