Collaborative Problem Solving using an Open Modeling Environment

C. Fidas[1], V. Komis[1], N.M. Avouris[1], A Dimitracopoulou[2]
[1]University of Patras, Patras, Greece
[2] University of the Aegean, Rhodes, Greece
fidas@clab.ee.upatras.gr, komis@upatras.gr, N.Avouris@ee.upatras.gr,
adimitr@Rhodes.Aegean.gr

ABSTRACT
This paper presents ModelsCreator version 3 (MC3) an environment that supports collaborative building of various kinds of models. MC3 is an environment that permits synchronous interaction of students at a distance who collaborate in building models out of primitive objects. The open character of MC3 means that students have access to an open set of primitive objects 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 objects, in order to obtain a solution. In this paper we concentrate on the architecture of MC3 and the basic functionality of the environment. In particular we study the effect of the open character of the environment in collaboration and problem solving. Through experimentation it is demonstrated that the users of MC3 can have rich interaction in order to build or exchange at run time the necessary primitive objects for model building.

Keywords
Computer supported collaborative learning, open learning environments, computer supported collaboration, semi-quantitative modeling

INTRODUCTION
The importance of modeling of phenomena, activities or systems in learning has been widely recognized (Bliss, 1994). A number of software tools have been developed during the last years that support learning through modeling. These software environments mostly concern mathematical models of physical phenomena (Teodoro, 1997), while other modeling activities have also been proposed, like creation of concept maps, test of logical propositions, modeling of ecological and other complex phenomena (Soloway et al. 1996), etc. A special case of modeling tools and activities relate to semi-quantitative modeling, proposed by scientists from science education and psychology fields (Bliss 1994). Their purpose is to support children's reasoning and help them have access to quantitative reasoning in a progressive way (Ogborn, 1998). Semi-quantitative models may involve countable concepts, however they do not reflect their values. The reasoning of the students engaged in semi-quantitative modeling involves studying in a complex system how the approximate value of a concept or object property has an effect on other properties, which may in turn, affect other parts of the model.

MODELSCREATOR (MC) is a modeling learning environment, under development and experimental use during the last few years (Dimitracopoulou et al. 1999, Komis et al. 2001). MC supports expression of different kinds of models mostly for students 11-16 years old. It integrates dynamic models: semi-quantitative models, quantitative models, and executable decision making models as well as static qualitative models (concept maps), with special emphasis on semi-quantitative modeling. These models meet the requirements of many curriculum subject matters, permitting interdisciplinary use of the modeling process. MC puts great emphasis on visualization of the modeling entities, their properties and their relations. Visualization is crucial in supporting the reasoning development of young students and favors the transition from reasoning over objects to reasoning with abstract concepts (Teodoro 1997). This feature is extended also to the simulation of executable models allowing their validation through representation of the phenomenon itself in a visual way and not in an abstract one, as it is usually the case.

The most recent development of ModelsCreator (MC version 3.0, MC3), reported here, has been in two directions. One concerns its transformation to an open modeling system and the second its support for synchronous and asynchronous collaborative development of models by distant groups of young students. This paper focuses on presentation of these new features, in terms of system architecture and functionality and the implications of their inter-relation.

The importance of the open character of the MC3 environment on collaborative modeling is discussed first. In a typical closed collaborative problem-solving environment the students have at their disposal a common set of basic

constitutive elements, out of which they construct their representations or the jointly developed models. These primitives can be rectangles, ellipses, squares, different statement types, etc., as it is the case in Belvedere (Suthers and Jones 1997), COLER (Constantino and Suthers, 2001), C-CHENE (Baker and Lund, 1997), Modeler Tool (Koch et all 2001). So common understanding is based on the existence of these common basic primitives. In contrary, in an open learning system like MC3, one student before entering in a specific collaborative session may build individually a new set of primitive elements to which meaning can be assigned. The student is provided with adequate tools (editors) that permit creation of these new entities or modification of existing ones. As a consequence, collaborating students may find themselves in possession of heterogeneous sets of primitive objects. Also compound primitive objects, like partial models can be built that can be associated with the new ones in multiple modes. Even if the collaborating partners share a problem definition and given data set, one or more of the partners may have access to additional basic constructs or compound primitives, making the process of grounding of interaction and common understanding particularly complex. These open collaboration environments, as they become available, set new challenges in collaborative problem solving, necessitating new functionalities, (Dillenbourg *et al*., 1995, Muehlenbrock *et al*., 1998) and eventually resulting in more semantically rich patterns of interaction and grounding mechanisms (Baker et all 2001).

This paper focuses on the architecture and the main functionality of the open collaborative MC3. We present the characteristics of the environment that make synchronous model building possible. Special emphasis is put in the collaboration protocols supported and in the new objects creation functionality. Extracts of interaction during problem solving are included in this paper that demonstrate some effects of the heterogeneous concept libraries on interaction and problem solving.

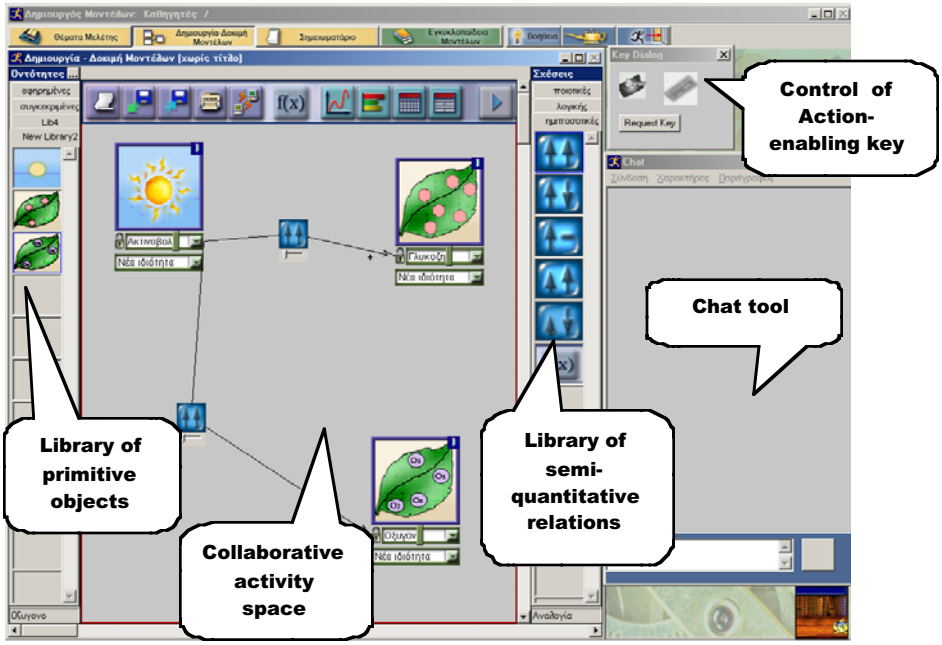

Figure 1. The ModelsCreator 2 User Interface during model building

COLLABORATIVE MODELING WITH MODELSCREATOR 3.

MC3 permits the collaborative building, testing and validation of models. The main functionality of the environment relates to the Activity Space where the models can be built, shown in figure 1. This space contains tools necessary to construct models, tools to represent models in alternative ways and tools that can run the models. In order to design a model, students have to insert primitive entities, set their properties and create relations between them. A part of the library of available primitive entities is shown on the left of the activity space, in figure 1, while on the right there is the list of available (semi-quantitative) relations. These relations link concrete objects properties or concepts and express variation of properties' values and direction of this variation. The relations are represented through symbols. For instance, the relations of analogy or inverse analogy are represented by the symbols:__, __, (see figure 1) expressing reasoning such as: "If one entity increases, the other one might increase, or decrease". The students can use a variety of simple relations that correspond to hidden algebraic formulas. The entities can represent concrete concepts with iconic representation that can change as the properties of the object change (e.g. the amount of water in a container or the amount of oxygen participating in photosynthesis). They can

also represent abstract concepts with mostly textual or iconic representation. For each entity, one or more properties have to be determined. During model testing the properties' values affect the appearance of the objects introduced in the activity space, representing the concepts. For instance, the volume of the water (variable) in a container (object) affects the representation of the container. The students can link entities experimenting with the available semi-quantitative relations.

MC3 can be used either as a stand-alone learning environment, or as a collaborative modeling environment. The latter is presented in more detail here. The collaboration is enabled both through asynchronous and synchronous interaction of distance partners. The integrated **chat facility**, shown on the right of figure 1, permits exchange of free-text messages between collaborating partners. Also a synchronous and asynchronous **object exchange tool** has been implemented. If the recipient of a compound or a primitive object is on-line during transmission, the object is sent directly to the receiving partner. If the recipient is off-line, the model or object is stored in an ftp-server and when the recipient is connected, the transmission is completed, as discussed in the system architecture section bellow.

The **Activity Space** can become a drawing space of synchronous collaboration, in which one of the two collaborating partners can insert primary objects (concepts and relations), 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 activity space. These actions generate messages transmitted to the passive partner, thus reproducing the same effect at the screen of both workstations. So MC3 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. The key-passing tool is shown on the top right corner of figure 1. An implication of this "key exchange" protocol is that deadlocks can be created in cases when the active 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 experiments. 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 consideration seems to be in agreement with the view expressed by researchers of similar environments, see (Soller, 2001).

Variations and enhancements of the above standard protocol involve firstly the possibility of creation of "invisible components" by one of the partners, thus modifying the semantics of the WYSIWIS environment, secondly a mechanism has been developed for interleaving text messages and action by a deictic tool that has taken the form of *sticky notes* in the activity space (see Fidas *et al*. 2001). Additionally alternative protocols for controlling *ownership of parts of the model* have been devised, so that collaborating partner cannot modify parts of the solution that have been built by another partner. In the last section of the paper an experiment studying the effect of variation of solution ownership protocol on problem solving is described

A component of MC3 that contributes to its open character is the **Editor of primitive objects**. The students or the teachers can define primitive objects and insert them in their object libraries or in public repositories. Properties are assigned to these objects and iconic representations that correspond to range of values of the defined properties. Also hidden functions can be defined that interrelate the properties of the object, thus providing it with "behavior". In figure 3, editing of property "*Water*" of object *Plant* in a photosynthesis library is shown using this Editor. The implication is that the libraries of objects in possession of collaborating students can differ, necessitating new patterns of interaction during problem solving, as discussed in the following sections.

An underlying property of the Editor relates to the global unique identifier (GUID) given to each new object built, so that different objects can be distinguished in a common repository. The approach followed in MC3 is that objects receive unique identities locally upon creation without the support of a central broker: Every object which is built is allocated a unique GUID by an algorithm which produces a 128 – bit number based on the IP address, network card number and the local time and date of the host. The same algorithm has been previously used with success to identify global and unique software components.
In the following section a more technical description of MC3 is provided, outlining the design of MC3 that implements the described functionality.

.

Figure 3. Primitive Object Editor

MODELSCREATOR 3.O ARCHITECTURE

In an open collaborative modeling environment the two main functionalities of the system relate with (a) building and sharing new primitive objects and (b) collaboration support in the presence of heterogeneous libraries of primitive objects. In this section an insight into the architecture of MC3 collaborative system is given. An overview of the system architecture is provided in figure 4, presenting the main functionalities of the system described in this paper. A modular approach has been followed, in order to reduce the complexity of the design. The aim of each module is to provide specific services to the modules with which it is connected, isolating the details of the construction of these services.

In figure 4 two collaborating host installations of MC3 and their Server are shown. The Server functionality is to manage the users' accounts, to provide database facilities for objects or models for asynchronous communication among the users. Also a facility of web-based search of public object repositories through the *Http Sever* is enabled. While full specification of this architecture is beyond the scope of this paper, some typical interaction scenarios are described in this section that illuminate the functionality of the architecture.

(1) User Login

1a) User login information is sent from the User Host to the Broker *Host Interaction Agent* (HIA)

1b) The HIA informs the user's database on the Server about the user login.

1c) Upon login, a check is performed if there are any objects to be dispatched to the user .If there are any, they are sent through the *ftp modules*. (1d,1e).

(2) Exchange of objects and models

The exchange tool is a synchronous/ asynchronous communication tool. If the recipient of the object or model is on-line during the transmission the file is sent directly to the receiving partner through the hosts *ftp-modules* (2a). If the recipient is off-line the ftp module of Host A sends the file and the user-id of the recipient, to the *Broker ftp module* (2b). Subsequently, the Broker ftp module stores the file and updates the User's Database (2c). The next time the recipient is logged in the transmission is completed, as discussed in (1).

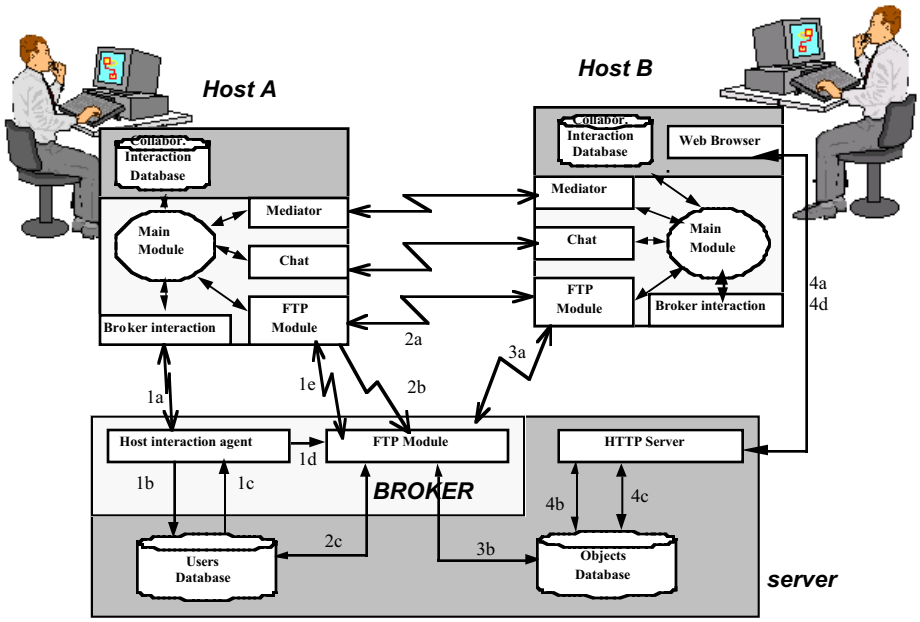

Figure 4. The MC3 system architecture

(3) Object publishing

New objects can be published to commonly available repositories. The key to the attainment of an open environment is the separation of the educational material from the environment. In the MC3 architecture each object is a *file*, which carries all the information needed in order to be used in the MC3 environment. Interoperability is achieved through the fact that each published object supports a standard COM interface and therefore it can be imported in every application which supports this standard. Moreover each object has in its heading a meta-level description containing keywords which describe the object in a conceptual way and can be easily searched by the users, as described in case (4).

This case describes the scenario of explicit publishing of primitive objects or entire models by a user in the Broker database. This permits sharing of objects by communities of students and teachers.

3a) The *FTP Module* of Host B sends the file to the *Broker FTP Module*. L

3b) The Broker FTP Module stores the object and updates the object's database about the new object. The files are stored in folders named after the object Global Unique Identifier.

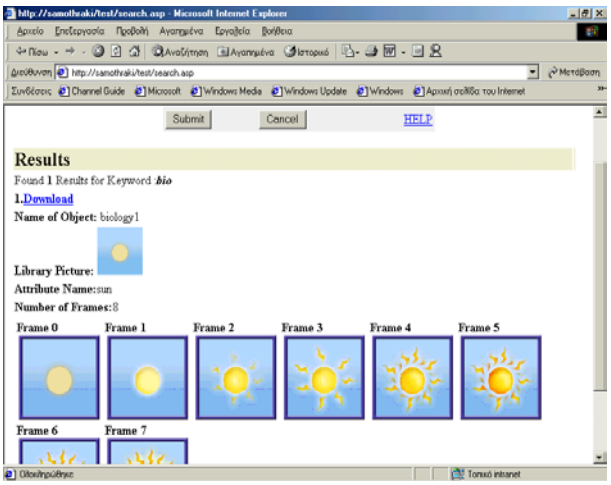

Figure 5. Web interface to search published material

(4) Search in published material

The created and published material can be searched through a web interface. The user sends through the browser an HTPP request and the keywords, which describe conceptually the searched objects (4a) to the http server. After searching in the Object's Database (4b) for objects which best address the users needs, the results are send back to the user (4d). In figure 5 a primitive object is shown as a search result. Eight (8) frames, corresponding to different values of the attribute *sun* can be seen in this figure.

(5) Logging of interactions

History of interaction during problem solving can be stored in a database, located in the student host, see figure 4. The log files contain actions in the activity space and exchanged text messages. These are organized according to the user, the type of interaction (e.g. proposition, insertion, rejection), the primitive object involved and the solution (the produced model). Having the interaction stored in a database means that it is easy to obtain structured information on interaction through appropriate queries. An analysis model (Avouris *et al*. 2001) has inspired the structure of this database. In figure 6 a typical view over interaction logging data is shown. In this picture, one can see that hyperlinks relate different views of interaction data. The capability of the environment to log all user activity in the Activity space together with the synchronous/ asynchronous communication actions, makes MC3 a powerful testebed for collaborative problem solving research, as described in (Komis *et al*., 2001).

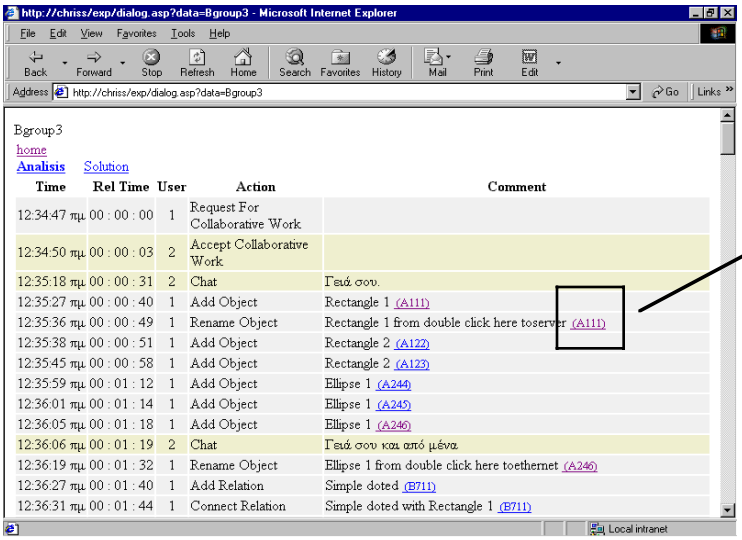

Figure 6. Logging of interaction data presentation

(6) Synchronization of Shared Activity Space through the Mediators

Most of the existing CSCL systems achieve synchronization among the students' activity spaces according to a server-client model (replicated architectures). The server monitors the status of the shared activity space and manages its consistency by informing the clients about the changes through message passing.

According to our approach synchronization is achieved using a *peer-to-peer protocol*, without intervention of a 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. 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**, shown in figure 4. The size of these 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.

(7) Collaboration of users with heterogeneous libraries of objects

The previously discussed case (6) 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 MC3 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 GUID. Mediator B searches the local Object 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 (t1). Mediator A sends the object, through its *ftp module*, and waits (t2). 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.

CASE STUDY OF COLLABORATIVE PROBLEM SOLVING

In the previous sections the main functionality and architecture of MC3 has been described. In this section experimental use of the developed prototype is presented. Two experiments are briefly presented and discussed. They concern usability and cognitive questions in relation to introduction and exchange of new entities during collaboration and ownership control of newly introduced entities in the activity space.

Study (I) The objective of this experiment was to monitor interaction of students in the case of heterogeneous libraries of primitive objects thus studying the effect of the open architecture on problem solving. The experiment took place in the frame of the graduate course of the Early Childhood Education Department, where four graduate students were asked to study and model collaboratively the factors affecting plant growth using MC3. Various primitive objects were made available relating to the photosynthesis process. Two students collaborated in each experiment. In case (A) some key primitive objects were missing in one of the two partners' library, while in case (B) key objects were missing from both students' libraries.

A general conclusion was that in both cases the groups managed collaboratively to produce a meaningful solution. In both cases there was rich interaction on availability of necessary primitive objects and both groups managed to identify missing key objects. In the case of group (A) there have been incidents in which the active partner failed to identify a primitive object and asked the partner through the chat tool to look for it in her library. Once the object was identified by the partner, the key possession has changed in order for the owner of the primitive object to insert it in the activity space. In the case of group (B) in two cases the partners decided that a necessary object was missing and they proceeded with building it at run time and incorporate it in their joined model. Extracts of interaction and the produced solutions are included in the following.

Table 1. Extract of interaction between partners of Group (A)

1	User1 :Chat :now we need some water
2	User2 :Chat :DO YOU HAVE SOME OBJECT LIKE WATER?
3	User1 :Chat :I look now
4	User1 :Chat :yes I have a spring
5	User2 :Chat :ASK FOR THE KEY AND INTRODUCE IT PLEASE
6	User1 :Request key
7	User2 :Key Request Accepted
8	User1 :Insert Object : Spring
9	User1 :Chat :what relation should I put between water and plant?
10	User2 :Chat :MORE WATER MEANS MORE GROW
11	User1 :Chat :ok

From this extract it is evident that user 2 is in search of an object that cannot find in the local library. The user takes the initiative to ask the partner to look for it providing some precise verbal description ("an object like water"). The action key is conceded to user1 in order to achieve the objective of introducing an object with the required property. It seems that there is an agreement on a common objective between the two partners and that the chat tool is extensively used in this extract in order to identify the required object.

In the case of group (B) there have been two incidents during which the partners ended in deadlock, after searching for a specific object. They agreed to build a new object and this was done at run time. An extract of interaction that lead to new primitive object creation is included in the Table 2. From Table 2 it is evident that User 1 is searching for an "object like rain". However despite the fact that this user has taken the initiative to search for an object of these characteristics and fails to find one in her library or the partner's library, proceeds with suggesting of building one to User2. This is perhaps due to the disparity of skills of the two partners. User 2 was more experienced user of the environment and this was known to User1. So there was no conflict on who would build the missing object. There has been only a comment by user1 on the aesthetics of the bitmap of the developed new object. (action14).

Table 2 Extract of interaction between partners of Group (B)

```
1       User1 :Chat :what we should add?
2       User2 :Chat :YOU SAID SOMETHING ABOUT RAIN
3       User1 :Chat :yes
4       User1 :Chat :some rain
5       User2 :Chat :DO YOU HAVE AN OBJECT LIKE RAIN?
6       User2 :Chat :NO
7       User1 :Chat :do you want to try creating one object like
        this?
8       User1 :Chat :ok
9       …. (User2 uses the editor )
10      User1 : New Object:C:\rain.obj
11      User2 : Insert Object:rain
12      User2 :Chat :I THINK IT IS VERY BEAUTIFUL
13      User2 :Attribute :rain
14      User1 :Chat :_ think you are as good as me at drawing :)
15      User1 :Chat :but we can imagine that this is rain ...
```

In Figure 7 the solution of group (B) is shown. In this figure the distinct style of the two newly introduced objects can easily be identified.

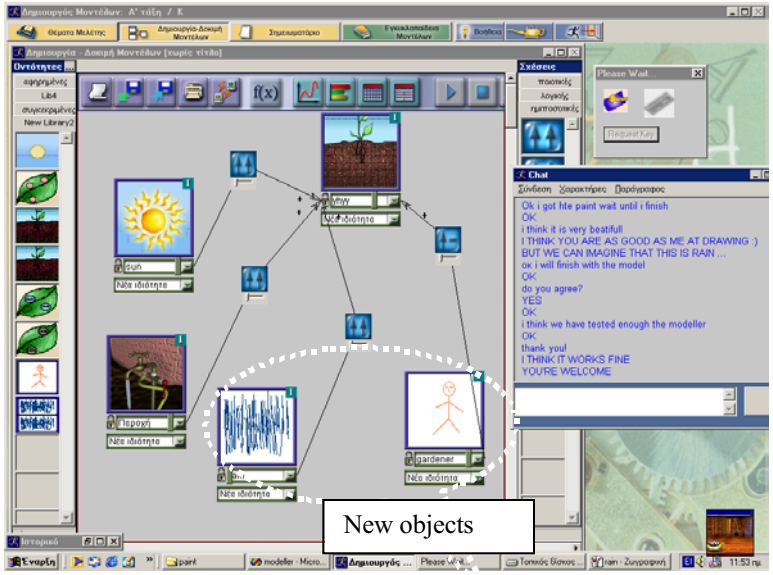

Figure 7. Group (B) solution

***Study* (II)** The second experiment involved 14 students of the Undergraduate course on Internet programming of the Electrical & Computer Engineering Department in the frame of a laboratory session. The main objective of this study was to evaluate the effect of solution ownership protocol variations on collaboration and interaction. Seven groups with similar characteristics were formed, collaborating in pairs. The problem-solving task involved the collaborative building of a diagrammatic model concerning the structure of the Internet. Two alternative collaboration protocols were used; Groups (A) had no ownership control while groups (B) maintained ownership of introduced objects, so partners were not allowed to modify objects introduced by their peers.

In the case of groups (B) every time a partner needed to modify an object of different ownership, a negotiation phase had to be initiated concerning the purpose of the modification, in order to convince the object owner on the proposed modification. This ownership control mechanism was effective in inhibiting modifications of created objects by other partners. 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 TO 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 Table 3 (group B4) partner 1 attempts to convince partner 2 to modify part of the solution:

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.

Table 3. Extract of interaction of group B4

55 : 35	1	Chat	You have connected two Routers; I think that you should connect the WANs.
56 : 27	2	*Chat*	You are right, but one does not need to link the two Bridges in order to affect the connection?
57 : 19	1	*Chat*	Right! However in the diagram we have grouped the networks in WANs so we have to do it through them!
57 : 47	2	*Chat*	OK I need the key to change them
57 : 51	2	*Request Key*	

CONCLUSIONS

An innovative environment enabling collaborative modeling activities has been introduced in this paper. ModelsCreator 3.0 (MC3) supports semi-quantitative, quantitative and qualitative reasoning during modeling activities of young children collaborating at a distance. MC3 presents innovative features briefly presented here. Synchronous modeling activity can be performed at a distance using MC3, based on a mechanism of light multiple processes (reactive agents) in collaborating hosts.

MC3 is an open modeling system since it provides the possibility through specific editors and open libraries to create new entities with various properties and behavior as well as new compound entities, models and problems. The MC3 system architecture and functionalities that enable this open character have been presented and discussed here: (i) A *repository* of publicly available modeling entities has been created and made available to the learners community in a common Server, (ii) *Search mechanisms* and web-based interface to this repository has also been developed, (iii) provision has been made to support unique identities (GUID) of any developed object (entities, models, problems) at the local host level, and entities exchange mechanisms have been established (iv) user protocols of collaboration for synchronization of shared activity space and online update of users' heterogeneous libraries have also been defined.

As a result, a software environment has been built, implementing the proposed architecture. This environment presents many interesting new features that need to be extensively evaluated. The first phase of this evaluation, reported briefly here, involved experimentation with specific functionalities. One experiment studied the effect of heterogeneous or missing primitive object libraries in problem solving. The result of this experiment was that the available functionality and tools allowed students to proceed with building models by collaboratively searching for missing primitive objects or develop new ones at run time, when required. One remark relating to this experiment concerns the extensive use of text-based messaging tools in this cognitively demanding activity. A limitation of the reported study is that the students that participated in this experiment were graduate students who could use the chat tool effectively and had developed typing and language skills. Something yet to be proven is the effect of this new degree of complexity on students of the target age group of this modeling tool, that is young students (age 11-16). A second experiment involved variations on the *solution ownership protocols*. It was proven that even slight variations of the developed interaction protocols affect the use of the tools and pose new demands in terms of cognitive tasks requested by the users. More experiments and investigations are currently planned, exploring grounding mechanisms during individual and collaborative construction of new primitives (entities and sub-models) for problem solving and modeling in sciences. The creation and use of these constructed primitives during various collaborative modes constitutes also a research direction for our team. Additionally an extended large scale use by learners communities of five European countries is planned in the frame of a new European project.

In conclusion, it seems that this new generation of complex collaboration support environments like open MC3, provides us with new enhanced capabilities and collaborative situations to be studied, creating a new degree of complexity in computer-supported collaborative problem solving. Their effective investigation poses new challenges to our research community and eventually necessitates new advances of the theoretical foundation of the field.

ACKNOWLEDGMENTS

Financial support has been provided by the IST2000/ModelingSpace Project of the European Union and the ModelsCreator/Pinelopi Program of the Greek Ministry of Education, under which the presented software environment has been developed. Special thanks are also due to our students who assisted and participated in the presented experiments.

REFERENCES

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

Baker M.J. & Lund K. (1997). Promoting reflective interactions in a computer –supported collaborative learning environment. *Journal in Computer Assisted Learning,* 13, 175-193.

Baker M.J., de Vries E., Lund K. & Quignard M (2001) Computer Epistemic Interactions for co-constructing scientific notions: Lessons Learned from a five-years research program.Proc. 1st EuroCSCL 2001, pp.89-96.

Bliss J. (1994). From Mental Models to Modelling in H. Mellar, J. Bliss, R. Boohan, J. Ogborn, C. Tompsett (Eds). *Learning with Artificial Worlds: Computer Based Modelling in the Curriculum,* The Falmer Press, London.

Constantino-Conzalez & Suthers D. (2001). Coaching Collaboration by Comparing Solutions and Tracking Participation. 1st EuroCSCL 2001, pp.173-180.

Dillenbourg P., Baker M., Blaye A., O'Malley C. (1995). The evolution of research on collaborative learning. In Spada E. & Reiman P. (Eds), *Learning Human and Machine: Towards an interdisciplinary learning science*, pp. 189-211, Oxford: Elsevier.

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.

Fidas C., Komis V., Avouris N.M. (2001). Design of collaboration-support tools for group problem solving, *Proceedings PC HCI 2001*, December 2001, Patras, Greece.

Koch J.H. Schlichter J. & Trondle P (2001). Munics: Modeling the flow of Information in Organisation. 1st EuroCSCL 2001, pp.348-355.

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.

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.

Ogborn J. (1998). Cognitive development and qualitative modeling, *Journal of Computer Assisted Learning,* 14, 292–307.

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

Soloway, E., Shari, L. and Jaskson, S. (1996), Learning Theory in Practice: Case Studies of learner- Centered Design, In Proc. *CHI 96. Human Factors in Computing Systems*, Vancouver, p. 189-196.

Suthers D. & Jones D. (1997), An Architecture for Intelligent Collaborative Educational Systems. In B. du Boulay, R. Mizoguchi (Eds) 8th World Conference on Artificial Intelligence in Education (AIED'97), pp.. 55-62.

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

Teodoro V.D. (1997). Modellus: Using a Computational Tool to Change the Teaching and Learning of Mathematics and Science, in "*New Technologies and the Role of the Teacher*" Open University, Milton Keynes, UK.

Teodoro V. D. (1994). Learning with Computer-Based Exploratory Environments in Science and Mathematics. in S. Vosniadou, E. De Corte, H. Mandl (Eds*.), Technology -Based Learning Environments : Psychological and Educational Foundations,* NATO ASI Series, Vol. 137, pp.179-186. Berlin : Springer Verlag.