Interactive Computer Aided Learning in the Didactic Activities of a Co-operative Virtual Classroom

Giuseppe Chiazzese[1], Maria Rita Laganà[2], Lucia Luminari[2], Elena Roberti [2]

1 Italian National Research Council - Institute for Educational and Training Technologies

2 University of Pisa - Department of Computer Science

Abstract

This paper highlights the educational strength of new technologies: merging interactive control techniques in co-operative environments can create ideal learning conditions. As an example, we will describe the didactic unit named “spelling dictation test”, supported by our network service based on the TCP/IP standard. Microsoft's Peedy agent is the teacher's assistant, which interacting with pupils through the Text-to Speech component, offers a real-time support while letting them doing their work at their own pace. The teacher chooses texts for dictation, which may be differentiated according to the learner's skills.

Key words: Education and Internet/Intranet Application, Computer based Learning, Multimedia Application.

1 Introduction

This paper describes a part of a wider project aiming to implement a co-operative environment: a “network classroom” made of pupils of homogeneous age ranging from 7 to 12 and their teachers. The members of this project have developed a communication protocol, based on TCP/IP standard, in order to support the "co-operative didactic activities" through a set of primitive actions, which represent the primary aspects of the learning process and the activities which take place in the classroom.[1][2].

Differently from similar environments, in our case just a single protocol summarizes the inner spirit of a real classroom: teachers enter and suggest discussion topics and activities, pupils may choose their seats and may interact (also send a secret note and wink at) each other. During test time, teachers can prevent pupils from talking to each other to improve concentration. Synchronous, asynchronous aspects coexist in the real and in the "virtual classroom" too. All of them are fundamental in learning activities, the former pertaining to real-time interaction (e.g. discussing with the teacher and class-mates) and the latter to afterthought (home-work, reflections about the lesson’s subjects). The virtual learning environment has been developed by taking into account the socio-constructivist theory which stresses the importance of a co-operative learning process [3][4].

The teacher has been equipped with the tools needed for being both a class-moderator and a knowledge-intermediary but not a supervisor. he/she can interrupt pupils' chattering but he/she is not allowed to read secret messages between students and cannot look at the personal pages of their copybooks.

Obviously, the use of a network has brought inherent advantages (e.g. the removal of space-time barriers) which coexist well in our application: “in spite of a choppy sea I can still attend the lesson, have a net-friend with a cultural background very different from mine).

With the intent of showing the capabilities of such a kind of environment we will describe a full working didactic unit that we have developed. The implementation of the errorless learning theory within a network has yielded ideal learning/teaching conditions: the teacher can be simultaneously close to any student, adjusting exercises to pupil’s skills and interests. This opportunity, would be totally unfeasible without a network; consequently, the implementation of the use of virtual classroom in LAN is strongly advisable.

Let us have a look now at the system architecture, its protocol and interface. Successively, we will give a description of didactic activities and a detailed view of the spelling dictation test.

2 System Architecture

The system architecture is based upon a client/server environment (fig. 1), within which there are operators with differentiated privileges (administrator, teacher, pupils). Client and server applications interact through a new protocol prototype, properly made for the requirements of network co-operative didactic environment [4].

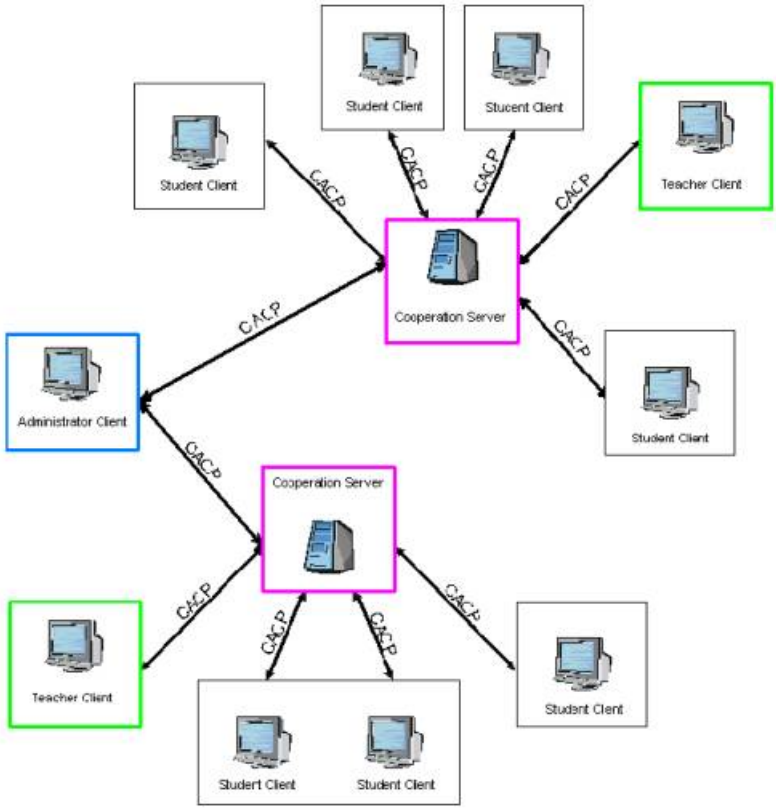

Figure 1

Specifically, the co-operation server is the supplier of all the services provided for the virtual classroom, and it also contains the database needed for the management of any didactic activity.

The server also dynamically, updates the information regarding the status; this leads to a correct management of the different virtual environments: classroom, home, common room and all the events occurring inside them: single, co-operative and playtime activities.

We define a "session" of our didactic system a group of attendants (possibly just one) interacting one another (or, possibly, just with the server) in either a synchronous or asynchronous way.

Different modes are available:

- *Individual Student Session*: students do their homework using the system individually. In this case, they can recover dialogues and discussions held during previous classroom sessions, also, they can look at the assigned homework, recover the resources stored in their schoolbags during their daily school activities.
- *Individual Teacher's Session*: teachers can enter the environment without their pupils presence, preparing, revising and correcting didactic activities;
- *Common Room Section*: pupils meet, without their teacher's presence, and carry on co-operative activities;
- *Ordinary Session*: pupils and teacher meet in the classroom carrying on regular didactic activities;
- *Secretary Session* the administrator manage the secretary activities such as enrolling students and assigning them to different classes, giving out timetables, etc. As it is shown in figure 1, the administrator can connect himself to different servers. Pupils and teachers get properly assigned to different classes: they receive a username, and passwords and profiles are recorded. From now on, each user is allowed to enter the virtual classroom.

The data needed to start the session are shown in figure 2: Besides identification Data, users can choose the kind of session they want to start.

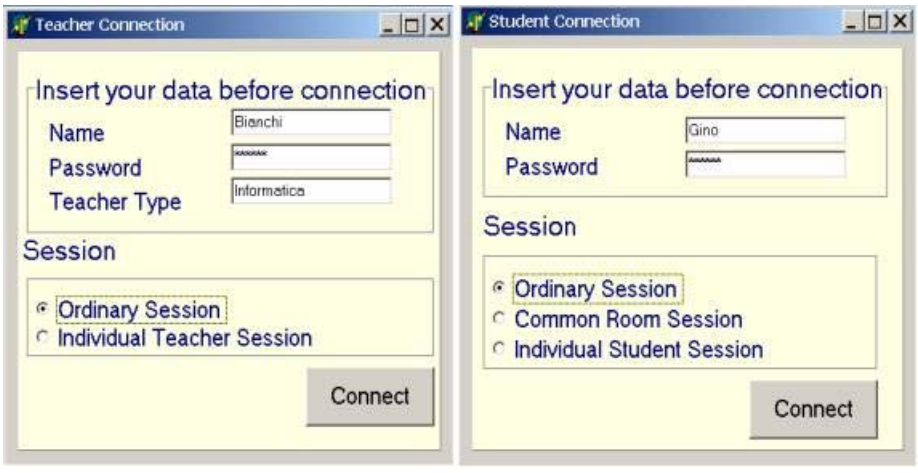

Figure 2

After the connection, both pupils and teachers may use the tools available. Modules for teachers and students have peculiar characteristics which are often of a dual type (asking for and granting to speak, sending and withdrawing a didactic unit).Therefore, they have a similar but not identical interface.

Due to flexibility and modularity reasons, our application has been developed into two layers: a communication protocol layer and a user interface one.

A description of the core of the application will be given in the next session which will explain our work better.

3 The core of the application: the CACP protocol

The set-up stage of our protocol requested an overall analysis of the dynamics which characterize a real classroom. We aimed to introduce a set of primitive functions emulating those social rules that grant for a co-operative learning during a didactic session [5]. While various communication systems (videoconference, chat, exchange of resources ...) already existed, a didactic system had to be built from scratch. None of the most common chat systems can emulate a dialogue in the classroom: therefore we have created a new displaying system which maintains the correct order of the written text,. Also, none of the available network services offers all functionalities under the same “umbrella”. Such an integration was not a programming exercise, instead it aimed to overcome the limits of the above systems: e.g. during a videoconference session both teacher and pupils behave differently from a live lesson. Pupils may only listen to the teacher (possibly listing their questions) but they are not allowed to ask their mates for help. On the other hand, teachers cannot vary the subject of their lessons by looking the attention into their pupils’ eyes.

The Co-operative Activities Control Protocol (CACP) functions allow for a full management of the various events which take place in the network session and make them available to the different applications (ClientPupil, ClientTeacher and Administrator). Communication primitives can be subdivided into four categories: content management, participants management, classroom interactions management, didactic activities selection and distribution [6][7]. Here, we give a brief description of some of them, while a detailed description of the networked dictation test primitives will be given in section 4.

The primitives of the contents management allow teachers to store didactic material onto the virtual classroom server. Such material includes homework, written tests, class’ dialogues, dictations tests.

The following primitives are related with the sending and the receiving of resources .

- GetResourceFrom
- PutResourceTo

The following primitives of the participant management allow the registration to the classroom

- RegistrationStudent
- RegistrationTeacher

A session is managed by

- CreateSession
- CloseSession

and by the subsequent connection/disconnection primitives

- JoiningSession
- LeavingSession

For the classroom interaction we show the following primitives. They improve the quality of textual communication implementing a new modality of messages indentation:

- MsgToAll
- MsgToChat
- MsgDifferentToChat
- DiscussionStateRequest
- DiscussionState

The first three (Msg....) primitives contain the parameters needed to locate the junction of the message tree to which the user's message must be attached. The remaining primitives enable to request and to return the whole dialogue should someone get connected in the middle of a session.

The idea of this visualization mode arises from the discussion lists, but its improved by the synchronous implementation [8].

Further, there are primitives which regulate discussions, allowing students to raise and to put down their hands and allowing teachers to enable student to talk

- PuttingUpHand
- PuttingDownHand
- GiveSpeech

Among the selection and distribution of activities there is:

- SendExercise

which has optional parameters (on/off) managing the communication between class-mates and the time limit for exercises.

- SendSolution

allows students to send the contents of their notebooks to the teacher for correction

In the next section, we give a detailed description of the virtual classroom interface, paying attention in particular to the management of the dictation test.

4 Interface

In the ClientPupil and ClientTeacher interface there are three sections.

- A toolbar, as shown in figure 3, equipped with a series of buttons, each one associated with an icon reminding its function.

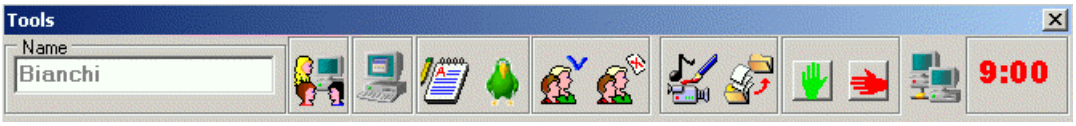

Figure 3

- a work area made of two distinct sections which are one dedicated to public discussions and another one for private discussion.
- another window which supplies for the spatial and temporal representation of the classroom through the use of icons. There are both teacher's and pupils desks. Those occupied bear the name of the students sitting there, so that new comers may "choose" their seat. The classroom composition is constantly updated, reporting the name of the students getting either connected or disconnected, the teachers entering and exiting the classroom, who is chatting, who is putting their hand up, and whether the discussion is public or private. Through this window, shown in figure 4, it is possible to send private messages and all activities to other class components.

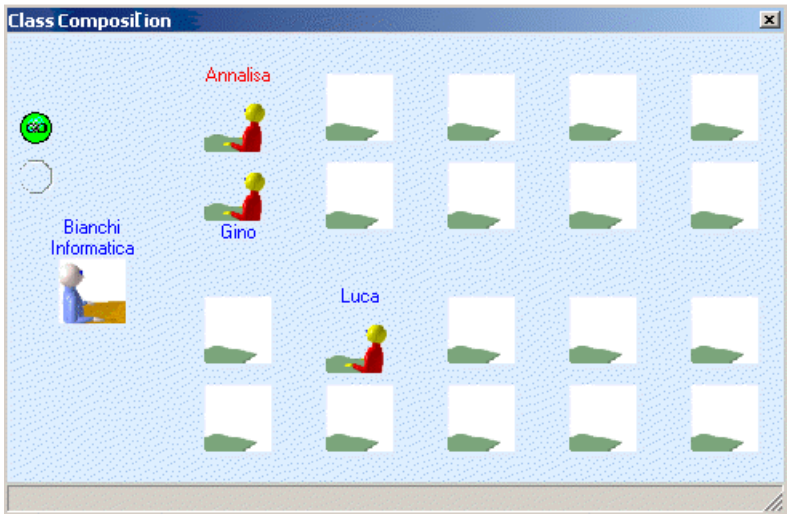

Figure 4

Each ClientPupil interface includes Arturo the parrot (Figure 5), which is the Microsoft's Peedy Agent, he interacts with pupils talking with them, giving voice to the teacher's suggestions. Arturo is a nice and engaging character who, through phrases and animations, attracts pupils attention, predisposing them to the didactic activity.

Today's text-to-speech engines can simulate voice in a sufficiently pleasant way, although they cannot reach the quality of a wave file. On the other hand, such engines do allow for a more versatile management of speech, which is put into direct correspondence with text arrays. Anyway, Arturo's resulting voice seems the "natural voice" of a parrot whose presence, we consider, was extremely purposeful.

Figure 5

5 Didactic activities

The didactic units give "life" to the co-operative classroom. By pressing the corresponding button teachers may use the didactic activities tools to assign them to pupils in a different way, depending on their interest and skills. We describe the following:

- Discussions. They are held within a specific area and may be "free" or "guided". In the second case the teacher allows the student to talk, in the first case the indented chat leads to a direct interaction[8].
- Exercises. Any kind of exercise may be selected, sent and downloaded, using the resources exchange tools.
- Cultural exchanges Resourse –exchange plays a fundamental role in co-operative learning, since it has several positive effects, like improving socialization between students and allowing for team-work. Learning of various regional slangs has been tested in Italy during a session held with pupils living in Tuscany and Sicily.[10]
- Controlled exercises. We have equipped our system with several didactic units: riddles, anagrams, synonyms and antonyms, word-translations, filling of uncompleted sentences, and spelling dictation test. These didactic units allow for an interactive check of pupils' behaviour, improving the learning process.

5.1 The spelling dictation test

We wanted to obtain a customisable system, tuned on the interests and requirements of each single participant. We reached this goal creating a strict synergy between teachers and software. The teachers were asked to make the most suitable choices for the students or as in this case the selection of the most appropriate text. The text must attract student's attention, for example it can be based on the child's favourite team or on the adventures of a hero. The software's task is to monitor activity in a friendly way. Therefore Arturo has replaced the teacher: he corrects mistakes in a friendly way rewarding when correct answers are given, and avoiding pupils from persisting in their mistakes. Arturo do not avoid a student from writing a word as the sequence of letters he/she prefers, but he avoids him/her from making a wrong association between visual and auditive memory. As soon as a mistake is made, a warning like "Pay attention, you wrote CASA while I said CASSA" is given. Our system compensates the lack of self-correction, displaying the right word. The educational importance of this solution is well known, since real-time feedback has always been considered as a vital component of didactic programmes.

The liveliness of children, for whom our system is thought of, has implied a series of problems that may be unnoticed at first sight. Let us imagine a child who changes previously checked words: to his own amusement! Especially if he knows that he will be rewarded with a "good job, son", while he/she has handed in a text full of mistakes. That is why we decided that the text that has been checked cannot be changed.

Every interaction is done without penalising or mortifying pupils, rather rewarding their achievements. Thanks to its adaptive characteristics, our system does not give rise to a forced pace to writing, allowing pupils to understand the word with his own time and typing skills.

Error checking may be done either "word by word" or "letter by letter". The former lets the pupils write an entire word before checking it. making a system able to understand the pupil's will is a complex problem. Our system is intended as a virtual representation of a real tutor checking the pupil's activity. While a human being can detect when a child waits for the next word, a virtual system needs complex decision processes. Surely a punctuation mark or a blank indicates where a word ends, but there are many other misleading conditions; e.g. a pupil may stop before typing a space, waiting for the next dictation word, or may type and endless word. In both cases, the system must interfere. Our system prevents the students from typing a word which is too long (in comparison with the correct one), and considers a word as finished when the lack of keyboard typing exceeds a certain amount of time.

We also mentioned the “letter by letter” option, this implies a more direct control, since warnings and suggestions are given each time a wrong letter is typed.

The teachers who are supporting us, consider the “letter by letter” option more suitable for mother-tongue dictations, while the “word by word” mode is more suitable for foreign language dictations. The goal is to increase the pupil's vocabulary independently from spelling errors. The two modes fulfil different didactic needs, and both lead to correct final results. In any case our system keeps track of errors, this will help teachers in identifying and correcting pupil-specific shortcomings.

The above two modes will be deeply tested when our system will be implemented at the school site “Istituto Comprensivo “Marco Tabarrini” di Pomarance – Pisa (Italy)”. This initiative has been promoted by Ms Stefania Ragoni, institute director, Mr Luigi Fantacci, chairman of the Centro Servizi di Volterra (Pisa), and the municipalities of Pomarance and Castelnuovo Val di Cecina.

5.2 Implementation of dictation test

Now we describe the methods of implementation of our system. A database of texts for dictations is stored on the server; texts may optionally be collected from web pages. They are sorted by language; Italian texts are further ranked upon their spelling difficulty, which are previously set by teachers during the input stage.

After choosing the dictation test, a teacher will be prompted with the interface shown in figure 6.

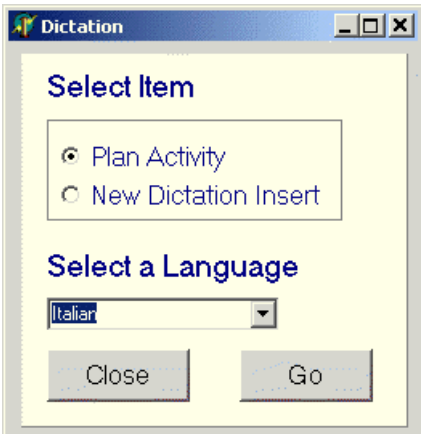

Figure 6

The “New Dictation Insert” option allows teachers to add a dictation text to the database. If an Italian text is added, the spelling difficulty option must be set too. We are pleased to notice that a lot of didactic material may be downloaded from web sites and catalogued upon teacher's personal criteria. For example in Italy, the Manuzio's project is creating a “portal of the primary Italian authors. The “Activity Plan” options are shown in figure 7.

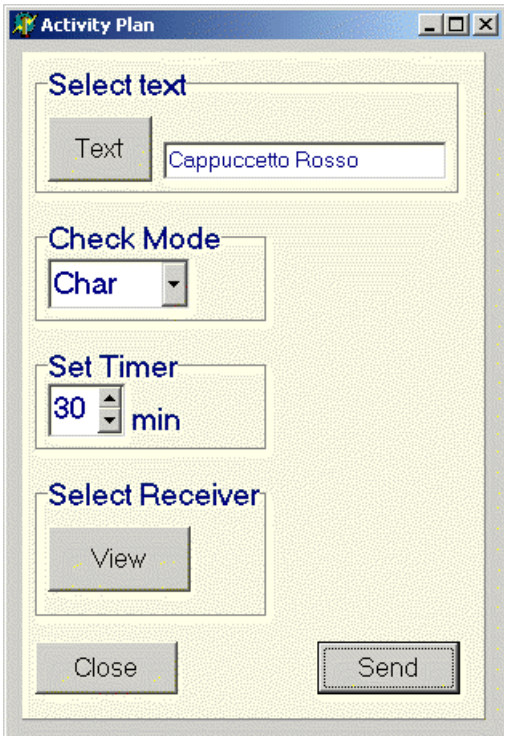

Figure 7

The "View" button activates the classroom-displaying window, from which the teacher selects the student who will receive the dictation. The "Text" button lets teachers access and select the dictations database. Both the errors checking mode and the time limit parameters must be set.

After a dictation has been sent, the notebook is opened with the parrot Arturo seated on it. After some preliminary animations, Arturo explains how the didactic activity has to be treated, inducing the pupil to start the exercise. In order to check the typed keys, we had to capture the typing event.. A key might be unintentionally pressed. In this case, our system does allow users to correct their mistakes. When an error is detected, the pupil is warned – You wrote "campana», I said "campagna». If the error persists, the right suggestion is given automatically. The time limit option allows for the system being tailored on pupil's skills. Slow-typing pupil's will receive suggestions, while faster ones will be able to type at their own speed. When handing in time elapses, the system will automatically complete unfinished exercises, sending them back to the server.

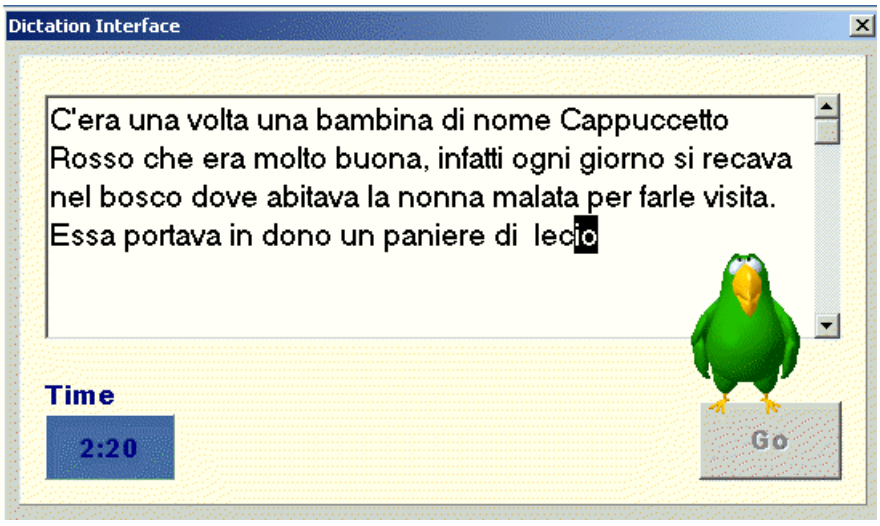

Figure 8

Dictation functionalities have been developed at a protocol level: they are functions for saving a text, selecting and collecting a text from the server, handing out a text to pupils, starting and stopping an activity.

We now list the primary primitives, stating the syntax of the protocol messages and their direction, and specifying sender and addressee. We employed the following convention:

TC = **T**eacher **C**lient, **CS** = **C**ooperation **S**erver, **SC** = **S**tudent **C**lient

The following messsages:

· RequestDictList?<> **TC->CS**

· ResponseDictationList?<DictList> **CS->TC**

allow teachers to interact with the server.

Text displaying is done by:

- DictationRead?<DictID> **TC->CS**
- Dictation?<DictText> **CS->TC**

while new dictation storage is done by:

- DictationWrite?<DictText> **TC->CS**
- OKDictationWrite?<> **CS->TC**

The primitive

- SendDictationTo?<DictID,DestList,Mode,MaxT> **TC->CS**

represents a teacher's request for sending the dictation DictID with the error-checking mode Mode and handing in time MaxT, to all the students listed in DestList. Upon such a request, the server sends to all DestList students the following primitive:

- Dictation?< Mode,MaxT,DictText> **CS->SC**

where DictTest is the dictation test retrieved from the server's memory upon identification DictID. Next, the teacher is sent the primitive:

- OKSendDictationTo?<> **CS->TC**

which confirms the activity dispatch. After the activity has ended, the ClientPupil sends the message:

- DictResult?<ErrorFile,MaxT,Time> **SC->CS**

This enable the activity results to be stored onto the server and the teacher being notified with:

- DictResult?<ErrorFile,MaxT,Time> **CS->TC**

Conclusions

The employment of new technologies may build ideal learning environments. Co-operative tools are created, allowing a pupil to be partnered by net-classmates and guided by a real, human tutor. Such a tutor is neither a today's Aristotle embodying the ancient Artificial Intelligence dream or a mechanic teacher administering a sterile education. Rather the teacher is the one we all know. What is new compared to education in a real classroom? Is it the space-barriers remission? No! although it is very important : each pupil has *interactive* real time guidance and the teacher in charge has the decision-making role. In fact the environment we propose, amplifies human exchanges (sharing an opinion, being next to someone while they are working on the task) which guarantee effective learning At the same time each student in the virtual classroom does his homework according to his own rhythm. The conditions created by an education-oriented cooperation environment are so innovative that they reveal their benefits in LAN within a real classroom.

Our next programming efforts will aim to increase the number of virtual classroom functions: e.g. letting a teacher wander between desks and sharing the pupils'desktop, again, letting a pupil go into the "near classroom" to confront common didactics activities.

The ClientAdministator role will be soon implemented, developing the primitives that manage the in-classroom registration and the promotion to higher forms.

At the same time we will test the system with the collaboration of the interested schools.

Acknowledgements

We warmly thank Daniele Ciuffardi and Linda Nocera for their in help in the translation of this paper into English.

References

[1] Chiazzese G.; Cortopassi C.; Laganà M.R.: Virtual secondary school classroom on the Net in *International Perspectives on Tele-Education and Virtual Learning Environments*, Graham Orange, Dave Hobbs Editors Orange G and Hobbs DJ (eds) (2000), Ashgate, ISBN 0 7546 1202 3;

[2] Allegra M.; Chiazzese G.; Laganà M.R.: *A multimedia system for tele-education in Secondary school*, ICL '99 Conferrence "Interactive Computer aided Learning - Tools and Application", Villach / Austria 7-8 October 1999

[3]Harel ,I.; Papert S.: *Constructionism*, Ablex Publishing Corporation, Norwood, New Jersey (1991);

[4] Allegra M.; Chifari A.; Fulantelli G.; Ottaviano S.: An On Line Cooperative Learning Environment, *Canadian Journal of Education Communication*, 2(26),1997,125-132

[5]Slavin,R.: *Cooperative learning*, Longman, New York (1983);

[6] Allegra M.; Chiazzese G.; Laganà M.R.: *An Internet service to develop cooperative learning environments*, IASTED, Internet and Multimedia System and Applications (IMSA '99);

[7] Schäll T.: Workflow Management Systems for Process Organizations, *LNCS, Vol 1096*, 1996;

[8] Allegra M., Chiazzese G., Laganà M.R., *Synchronous textual interaction in a virtual classroom: beyond chat* IASTED International Conference on Computer and Advanced Technology in Education, Cancun, Mexico, May 2000;

[9] Chiazzese G.; Chifari A.; Mannini A.; Ottaviano S.: *Una rete didattica cooperativa: studenti di Pisa e Palermo compagni di banco*, Informatica Didattica e Disabilità, Andria conference(IDD '99), 4-5-6 November 1999;

[10] Microsoft Corporation: *Microsoft Agent Software Development Kit*, Microsoft Press (1999);

Authors

Giuseppe, Chiazzese, Dott
Italian National Research Council- Institute for Educational and Training Technologies
Via Ugo La Malfa,153 - 90100 Palermo – Italy
chiazzese@itdf.pa.cnr.it

Maria Rita, Laganà, Assistant Professor
Department of Computer Science
University of Pisa
Corso Italia 40. 56100 Pisa – Italy
lagana@di.unipi.it

Lucia, Luminari, Student
Department of Computer Science
University of Pisa
Corso Italia 40. 56100 Pisa - Italy
luminari@di.unipi.it

Elena, Roberti, Student
Department of Computer Science
University of Pisa
Corso Italia 40. 56100 Pisa - Italy
roberti@di.unipi.it