A FRAMEWORK TO FACILITATE BUILDING OF COLLABORATIVE LEARNING APPLICATIONS

Meletis Margaritis, Christos Fidas and Nikolaos Avouris

Human-Computer Interaction Group, University of Patras, GR-26500 Rio Patras, Greece

margaritis@ee.upatras.gr, fidas@ee.upatras.gr, avouris@upatras.gr

Abstract

Design and development of computer supported collaborative learning systems is a complicated technological endeavour, since it involves tackling difficult distributed software design issues and adaptation to the continuously shifting technological background. In this paper a framework for the development of Computer Supported Collaborative Learning (CSCL) applications is proposed. The pedagogical value of the proposed framework is based on the fact that it allows the initiation of collaborative sessions with diverse settings, regarding the group size, the floor control and the level of peer awareness, thus facilitating teachers to design a variety of collaborative learning activities. Various evaluation studies have been performed using applications built according to the proposed framework and the obtained results are described.

Key Words: Middleware frameworks, groupware applications, replicated architectures, CSCL applications, collaboration services

1. Introduction

Modern teaching and learning practice often involves collaboration among students and tutors, encouraging construction of knowledge and building of meaning [1]. The main benefits of collaborative learning are related to the active character of the learning process, the deep level of information processing and the requirement of deep understanding from the students involved [2], [1]. Through such approaches, skills of critical thinking, communication and coordination can be developed and conscious knowledge construction mechanisms can be built [3]. Technology supporting synchronous or asynchronous collaboration plays an important role in this context. Network-based computer systems offer new possibilities for collaborative learning and at the same time raise new questions related to pedagogical and technological aspects. While the pedagogical point of view examines the feasibility and effectiveness of learning through collaboration related to factors like group synthesis, learning scenarios, cognitive transformation of the peers etc., from the technological point of view it is interesting to investigate new methods that facilitate the development process of Computer Supported Collaborative Learning (CSCL) applications.

The design and development of CSCL applications includes high complexity software design issues such as underlying architecture decisions, multiple users support, synchronization of tasks, consistency of data which are mainly distributed in different physical locations, session support, communication abstraction, coordination and awareness support, network bandwidth requirements and security issues. These decisions are related and affect the functionality and pedagogical affordances of the CSCL applications. So, development and maintenance of CSCL applications tends to be a difficult task, requiring high technical skills and great effort, which needs to be reproduced across different cases.

One possible solution to this problem is to provide abstract software constructs in the form of a middleware that offers a generic solution to these design problems and can be reused across different CSCL applications. This way the complexity of the design task is reduced and the shifting ground of the underlying technology is isolated from the CSCL application developers. This idea has been also proposed for the development of general purpose groupware systems. So, during the last years a number of middleware frameworks have been proposed with the aim to improve the development process of groupware applications.

Examples of such frameworks are Jade [4], Thyme [5] or GroupKit [6], briefly discussed next.

The *Jade* framework proposes a generic component framework that supports systems interoperability. However, this proposal focuses on inter-communication capabilities instead of collaborative services. Another interesting approach, bearing many similarities with the one included here, is the *Thyme* framework. The Thyme architecture is capable of creating replicated groupware applications. According to this approach a collection of components is provided which implements specific collaboration features such as communication via chat, shared whiteboards etc. GroupKit is a platform used in the development of replicated real-time groupware systems. By lacking modularity of component architecture, each GroupKit client is however strictly coupled with the application thus leaving many complex software design issues to the developer. Although these discussed frameworks provide solutions towards faster development of groupware applications they do not support an extensible and easy to use collaborative service infrastructure. Especially for CSCL systems it is important to have easy to use and install services that enhance communication and collaboration among the involved users in a cost and time effective manner.

In this paper we propose a middleware software framework, called the *"Abstract Collaborative Applications Building Framework"* (ACABF), aiming at facilitating the development of CSCL applications. ACABF provides services that turn a single-user application into a multi-user collaborative application, reducing development complexities that multi-user applications entail. Furthermore, ACABF that is constructed on top of a component groupware framework provides an abstract infrastructure offering a set of extensible collaborative services which cover numerous collaboration issues presented in the next section. The added value of ACABF lies in the fact that teachers are provided with the facility to design diverse computer mediated collaborative activities using the ACABF applications setting up many alternative collaboration parameters. This way they could have a benefit determining the optimal collaboration settings that would facilitate learning for a given context of use.

The paper is structured as follows: first the ACABF framework is presented, next the collaborative services of ACABF are briefly presented, followed by the interfaces between the services and the ACABF module. Further on, specific examples of CSCL applications that have been built using ACABF are presented: The *ModellingSpace* and the *Synergo* CSCL systems. Finally, a description of typical evaluation studies is given, providing evidence for the benefits of using applications built on the ACABF framework.

2. The ACABF framework

At the conceptual level, the *Abstract Collaborative Applications Building Framework (ACABF)* is designed to be used for providing essential *collaborative services* such as shared sessions, support for synchronous and asynchronous interactions, tackling of security issues, coordination mechanisms and awareness infrastructure. At the architectural level, it provides a middleware integration platform which provides interfaces that use protocols for the exchange of well structured messages among applications with the aim to provide collaboration facilities.

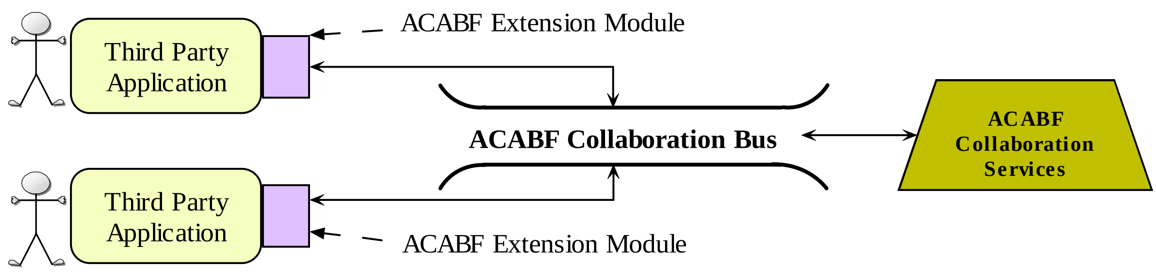

Figure 1. The communication model

We consider that the third party applications follow the *replicated architecture model*, i.e. each peer workstation maintains a local copy of the application process and shared workspace. Fig. 1 presents the ACABF role in such setting. In this figure one can see two peers interacting with a collaborative learning environment, for instance a physics modelling software. Through ACABF services the applications are synchronized and the peers can interact, share resources, etc. This way the standalone physics modelling software may be used as a CSCL environment for physics modelling.

Third party application can access the collaboration facilities of the proposed framework by embedding the "*ACABF Extension Module*", which is a built-in module that provides interfaces which implement communication and usage of the remote services by exchanging data through protocols for each provided service (fig. 2).

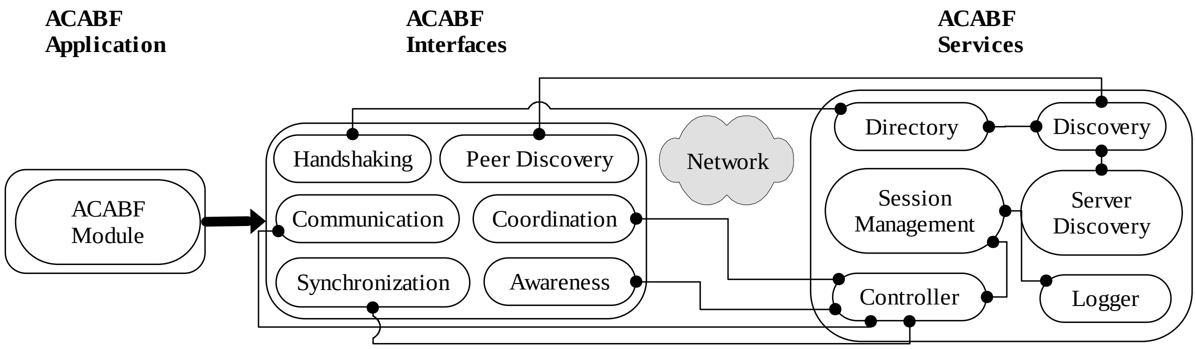

Figure 2. The ACABF framework overview

The communication protocol relies on a message-passing mechanism. The exchanged messages are structured following an XML format and each message is coupled to one interface. In the next section the conceptual presentation of the framework is discussed through the specific services that are supported.

3. The ACABF Collaboration Services

As it is shown in fig. 2, the framework is made of a number of services and interfaces. The services perform all the basic operations that are needed for a peer-to-peer collaboration, while the interface layers which are replicated and are part of the ACABF extension module provide the necessary protocol for the communication between an ACABF extension module and the ACABF services.

The ACABF framework includes six services, which are dedicated to provide different collaboration functionalities (direct communication, synchronization and coordination of peer workstations, awareness mechanism of the state of the peers and the group) to the users through the provided interfaces.

1. The *Directory Service* is responsible for keeping track of the peer clients currently connected to the ACABF framework. Since the proposed framework is a middleware framework, it provides its collaboration services to numerous third party applications. Thus, it is necessary to track presence of users for each application providing presence services to different communities. These services provide for every application a unique community of users, where different types of information are stored such as login data, workstations and status, nickname, IP address, availability, user profile, contacts etc.

The Directory Service interacts with the replicated applications through the "*Handshaking*" interface which introduces a third party application as a valid ACABF application providing all the information mentioned above. In this case the Directory Service supports this introduction of the new ACABF application instance and changes the status of the user in the directory.

2. The *Discovery Service* provides the users with awareness information about other users who are logged in to the community and are thus potential collaboration partners. It retrieves

and filters information coming from the Directory Service, concerning the online presence of users in the frame of a given community. The aim of this service is to separate users from different communities into different groups. This way various groups of users can have access to the collaboration services in an abstract manner.

3. The *Session Management Service* handls sessions of collaboration among participants of a specific application. A session is defined as a time frame in which a group of partners work towards achieving a common goal. A session starts at the time a user or a group of users accept the initial proposal for collaboration by the initiator performing an “acknowledge-reject” process with all the involving peers.

A collaborative session of peers might be continuous or may be interrupted. In the later case, the session service provides a facility that allows the peers to continue their collaboration activity in discrete time frames allowing thus a continuation of the collaboration activity. This service handles as well «late coming» support by allowing latecomers to join an ongoing session. The functionality of this service is implemented through invocation of session tracking information from the logger service, discussed next.

4. The *Logger Service* is responsible for keeping track of the synchronous or the asynchronous group activity. The structured messages produced and exchanged among the collaborating peers during a session are stored in the server and the peer workstations, as explained in the following. This service receives the data concerning the collaborative sessions from the Session Management service and saves this data in XML format. This data are important not just for re-establishing the activity after a breakdown or for supporting latecomers in a session but also for calculating various indicators of collaboration and for posterior analysis purposes of the collaborative activities.

5. The *Controller Service* is responsible for forwarding the messages among peers. Depending on the collaboration settings this service takes into account the information

provided by the Session Management Service in order to determine all possible receivers according to the coordination protocol (e.g. a private chat message should be forwarded to one recipient, a message for exchange of the floor control should be directed to the requester of the key). The coordination protocol defines the roles among the collaborative peers and the rights they have in modifying shared artefacts in common workspaces. An evaluation study, described in the case study section, has been performed to examine the effect of different coordination protocols in synchronous computer supported problem solving [7].

Furthermore, this service controls messages exchanged by the peers using the chat tool and the included dialogue openers. These dialogue openers are often used in collaborative environments for selecting the exchanged messages and providing structure to the dialogue, facilitating discourse analysis.

The peer awareness messages are also a part of messages controlled by the Controller Service. The awareness mechanism is the medium which provides the peers with the current status of their collaborating partners, as well as with the status of the whole group.

6. The *Server Discovery* service is used to discover other ACABF enabled local networks. This service can be useful for learning scenarios involving more that one schools, thus enabling inter-school collaboration.

5. Applying the framework in CSCL applications

The proposed framework has been already applied for building two CSCL applications with the aim to support groups of students engaged in collaborative activities. Although these applications are based on the same framework they show significant diversity in terms of application usage and supported learning activities. While Synergo (www.synergo.gr) supports synchronous collaboration in building diagrammatic representations (flowcharts, entity-relationship diagrams, concept maps, data flow diagrams), ModellingSpace environment supports building and exploring dynamic models with specified behaviour. A more detailed description of Synergo is provided in [8], while a thorough description of ModellingSpace is included in [9] while the pedagogical underpinnings of its use are discussed in [10]. Fig 3. shows a typical view of the Synergo environment during a collaborative session, demonstrating the use of the ACABF interfaces.

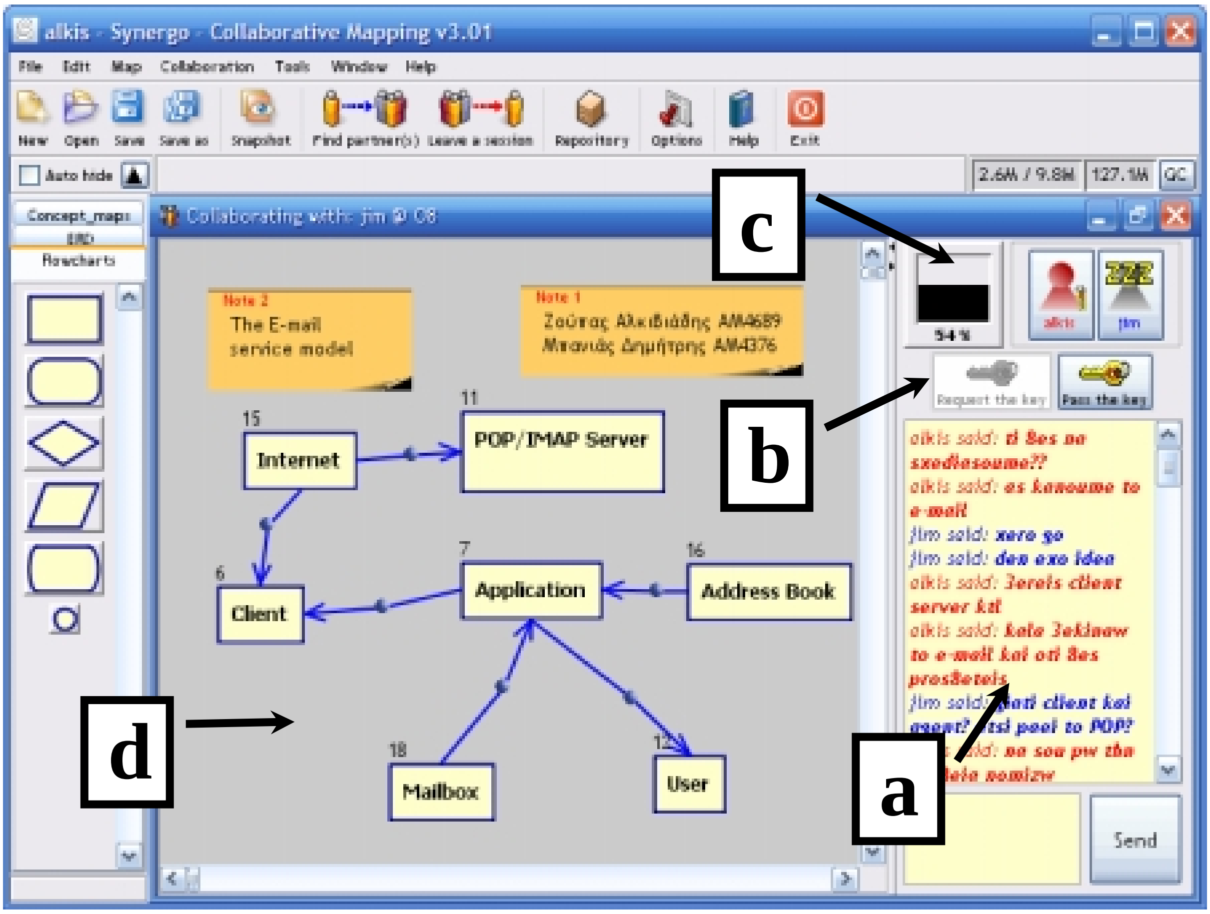

Figure 3. The Synergo main screen and the ACABF interfaces; a) the communication interface, b) the coordination interface, c) the awareness interface and d) the synchronization interface.

6. Evaluation studies

As described in the previous sections the added value of the proposed framework relies on the fact that it allows the parameterization of various collaboration settings thus enabling the examination of various collaboration parameters that affects collaborative learning. A general conclusions driven from several case studies performed is that the proposed system contributes to better learning through the support of collaboration among students, by providing collaboration services described above, thus encouraging the construction of knowledge and building of meaning [1],[2]. Furthermore, it has been observed that the semantics adopted by the proposed framework were easily understood by the students [11].

In the next sections two evaluation studies are presented which examine the effect of different group sizes and the effect of alternative floor control mechanisms in synchronous computer supported collaborative learning scenarios [11], [7].

6.1 The effect of group size

In reported studies of synchronous collaborative problem solving activities, there are mostly groups of two partners involved. In this study, we focused on groups of larger sizes and studied through an authentic educational activity the effect of group size on group performance and patterns of interaction. It was found that it is possible to design learning activities with group sizes greater than two and that the group size affects performance and interaction. It was discovered that group skill balance in particular is an important aspect in this context.

This study was contacted in the frame of a University course where the collaboration support architecture of ModellingSpace was used in the context of an authentic laboratory activity. Eighteen (18) students of the University of Patras, age of 22, during the laboratory of the undergraduate course on Internet Technology in the spring semester of 2003 were requested to build collaboratively, using a version of ModellingSpace, a concept map of an Internet service of their choice, out of those discussed in the class. They had one lab hour (around 45 min) available time to tackle the problem.

Six (6) groups of various sizes were built for this purpose. One (1) was made of four members, four (4) groups of three and one (1) of two members. The objective of this study was to investigate the effectiveness of the architecture and the provided tools to support groups of various sizes.

From this study it was found that in more balanced groups in terms of their members' skills, the participation of the group members in the activity is more balanced. In addition, the higher the size of the group, the more imbalanced the participation of the partners in the activity becomes. The group size was found to be a key factor for the quality of the solution (Correlation=0,664). The larger the group the more activity is produced. An additional observation was that the percentage of communication to action falls drastically with group size. So the larger the group, the higher percentage of activity is contributed to action than talk. Finally it was discovered that highly collaborative groups (i.e. groups in which the students have balanced activity) do not perform necessarily better in terms of academic achievement, an observation also made in other similar studies [12] [13].

6.2 The effect of different coordination mechanisms

As discussed previously, there is a possibility by ACABF to impose a model-level lock in the shared activity space for coordination of the activity through a token enabled floor control. Using ModellingSpace, an evaluation study was conducted in order to observe the effect of this coordination mechanism. The mechanism took the form of an Action Enabling Token, which is passed between the participants. This token owner is allowed to act in the shared workspace, while the rest just observe this activity and make comments through the chat tool. This mechanism is supported by token request function, which can be accepted or rejected by the token owner. Through this, the floor control is passed to other participants. Early experiments with this kind of floor control mechanism, see [7], have indicated that it may improve reasoning about action, as partners need to reason and negotiate when requesting the key. In the experiment reported here we have studied the effect of this mechanism on problem solving, by comparing the performance of two groups of students one of which used this mechanism while the other used no explicit floor control.

Thirty-two (32) students participated in the experiment in the frame of a scheduled class session. Sixteen (16) pairs of students with similar characteristics were formed. The collaborating pairs, dispersed in the computer lab, interacted for about 30', using ModellingSpace. Each pair of students was asked to produce, by the end of the session, a single solution to the problem, using the collaborative problem-solving environment. Eight pairs of students (group A) used the token-passing floor control mechanism, described in the previous section, while the other eight pairs had no explicit floor control mechanism imposed on them (group B).

The two groups produced solutions of similar quality to the given problem within the allocated time. So a first observation was that the existence or lack of the floor control mechanism did not seem to affect the performance of the students.

By studying in more detail the collaboration activity through the produced logfiles and solutions, it was observed that group B was more active than group A. The most important events per partner of group A were in average 109, while for group B 172. The difference was found statistically significant ($P=0.0131$). The complexity of the produced solutions by group B was also higher (it contained 27 objects against 21 in average), while this did not mean that the quality of solutions was necessarily higher.

7. Conclusions

In this paper a middleware framework for building computer supported collaborative learning systems has been presented. The framework aims at enhancing the development process of CSCL learning applications by providing collaboration services. The proposed framework has been successfully applied in two computer supported collaborative learning environments, *Synergo* and *ModellingSpace*, both developed using different pedagogical and research objectives. These environments have been used by communities of researchers and practitioners that gave valuable insights in the fundamental requirements for supporting small scale synchronous collaboration. As far as the pedagogical point of view is concerned, the proposed framework provides a variety of functionalities which can be used and parameterized by teachers, creating thus diverse educational scenarios in terms of group size, coordination protocol and level of awareness.

The evaluation studies performed, using applications built on ACABF, proved that a) the framework facilitates the building of collaborative application on various domains and b) the parametric nature of the ACABF framework facilitates the design of various collaborative scenarios with different collaboration settings. Furthermore, in studies performed by other researchers [14][15][16], dealing with large scale activities (involving more than 130 students in long term tasks), it was observed that the students involved understood the concepts of the coordination, communication and awareness mechanisms easily and accomplished successfully given tasks without facing major difficulties. The findings of those studies proved that students were motivated by the applications, which encouraged their collaborative participation and elicited particular forms of interaction among them, which could trigger the appropriate learning mechanisms [1].

Acknowledgements

The work reported here has been supported by the following projects: ModellingSpace IST-2000-25385, funded by the European Commission and Basic research program Herakleitos on Methods & Tools for CSCL, funded by the Hellenic Ministry of Education. Special thanks are due to Vassilis Komis, Angelique Dimitracopoulou and Vitor Theodoro for various suggestions and insights during the design of ModellingSpace and Synergo.

References

[1] P. Dillenbourg (Ed), *Collaborative Learning: Cognitive Computational Approaches* (Oxford: Elsevier, 1999) pp. 1-19

[2] M. Scardamalia & C. Bereiter, Computer Support for Knowledge – Building Communities. *The Journal of the Learning Sciences, 3*(3), 1994, 265-283.

[3] G. Stahl, Introduction: Foundations for a CSCL Community, *In G. Stahl, Computer Support For Collaborative Learning: Foundations For A CSCL Community*, *Proceeding of CSCL 2002*, Boulder, Colorado, USA, 2002, 1-2.

[4] M. Oliveira, J. Crowcroft & M. Slater, Component Framework Infrastructure for Virtual Environments, *Proceedings of the third international conference on Collaborative virtual environments*, San Francisco, California, United States, 2000, 139-146.

[5] S. Landsman, *The Tiny THYME, a manual for using the THYME framework*, Technical Report TR-02-231, Dept of Computer Science, Brandeis University, 2002.

[6] M. Roseman & S. Greenberg, Building Real Time Groupware with GroupKit, A Groupware Toolkit. *ACM Transactions on Computer Human Interaction, 3*(1), 1996, 66-106.

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

[8] N. Avouris, M. Margaritis & V. Komis, Modelling Interaction during small-group Synchronous problem solving activities: the Synergo approach, *2nd International Workshop on Designing Computational Models of Collaborative Learning Interaction, ITS 2004, 7th Conference on Intelligent Tutoring Systems*, Maceio, Brazil, 2004.

[9] N. Avouris, M. Margaritis, V. Komis, A. Saez, R. Melendez, ModellingSpace: Interaction Design and Architecture of a collaborative modelling environment, Proc. of 6th Conf. Computer Based Learning in Science (CBLIS), pp. 993-1004, Nicosia, Cyprus, 2003.

[10] A. Dimitracopoulou & V. Komis, Design principles for the support of modelling and collaboration in a technology based learning environment, *International Journal of Continuing Engineering Education and Lifelong Learning*, *15*(1/2), 2005, 30-55.

[11] N. Avouris, M. Margaritis and V. Komis, “The effect of group size in synchronous collaborative problem solving activities”, in *Proc. AACE ED-Media*, Lugano, 2004, pp. 4303-4306.

[12] V. Komis, N. Avouris, C. Fidas, (2002). Computer-supported collaborative concept mapping: Study of synchronous peer interaction, Education and Information Technologies vol.7, 2, pp.169-188.

[13] C. Fidas, V. Komis, S. Tzanavaris, and N. Avouris, Heterogeneity of learning material in synchronous computer-supported collaborative modelling, *Computers & Education, vol.44*(2), 2005, 135-154, 2005.

[14] Z. Smyrnaiou and A. Weil-Barais, Cognitive evaluation of a technology based learning environment for scientific education, *Proceedings of 6th International Conference CBLIS*, Nicosia, Cyprus, 2003, Vol . 2, 255-265.

[15] M. Xenos, N. Avouris, V. Komis, D. Stavrinoudis D & M. Margaritis, Synchronous Collaboration in Distance Education: A Case Study on a CS Course, *Proceedings of the 4th IEEE International Conference on Advance Learning Technologies*, Joensuu, Finland, 2004, 500-504.

[16] E. Voyiatzaki, C. Christakoudis, M. Margaritis & N. Avouris, Algorithms Teaching in Secondary Education: A collaborative Approach, *Proceedings of ED-Media 2004*, Lugano, Switzerland, 2004, 2781-2789.

Biographies

Meletis Margaritis, born in Patras, Greece (1977). Received Diploma in Electrical & Computer Engineering from University of Patras, Greece (2002). He is currently finalizing his PhD research in architectures for supporting computer supported collaboration of small groups in the same department and works as a senior researcher in several research projects. His main interests are in the areas of software technology with special interest in Human-Computer Interaction, in Computer Supported Collaborative Learning (CSCL) and Computer Supported Collaborative Work (CSCW) environments design and in Distance Learning. He is member of the Human-Computer Interaction Group of the University of Patras.

Christos Fidas, born in Munich, Germany (1976). Received Diploma in Electrical & Computer Engineering from University of Patras, Greece (2002) and PhD in Electrical & Computer Engineering from University of Patras, Greece (2004). His PhD thesis was in the area of design and development of computer supported collaborative learning applications. At the moment he serves as an Adjunct Professor of Computer Science at the Technical Education Institute, Patras Greece (2004-2006). His main interests are in the areas of software design and technology, groupware architectures and web services. He is member of the Human-Computer Interaction Group of the University of Patras.

Nikolaos Avouris, born in Zakynthos, Greece (1956). Received Diploma in Electrical Engineering from NTUA Greece (1979) and MSc (1981), PhD (1983) from the University of Manchester UMIST, UK. He served as a postdoc researcher at UMIST, UK (1983-1984), as Assistant Professor of Computer Science at the Technical Education Institute, Athens Greece (1985-1986), as Scientific Officer at the Joint Research Centre of the European Commission, at Ispra, Italy (1986-1993), as Software Engineer at the Public Power Corporation, Athens Greece (1993-1994) and then joined the University of Patras, to serve as Associate Professor (1994-2001) and Full Professor of Software Engineering and Human-Computer Interaction (2001- today). Founder and head of the Human-Computer Interaction Group (http://www.hci.ece.upatras.gr). Prof. Avouris main interests are related to design and evaluation of interactive systems, usability engineering, collaboration technology, context-aware computing systems and analysis and evaluation of collaborative activities.