Collaborative multimedia environment for chess teaching

Francisco J. García,* María N. Moreno and José R. García-Bermejo

Computer Science Department – Science Faculty,
University of Salamanca,
Plaza de los Caídos S/N,
Salamanca 37008, Spain
E-mail: fgarcia@usal.es
E-mail: mmg@usal.es
E-mail: coti@usal.es
*Corresponding author

Joaquín García

Theory and History of Education Department – Education Faculty,
University of Salamanca,
Campus Canalejas. Pº de Canalejas, 169,
Salamanca 37008, Spain
E-mail: carrasco@usal.es

Abstract: Chess teaching has not made extensive use of the so-called new technologies; in most cases, we just see a tutorial action in which e-mail is used to exchange moves that will be reproduced later by the student, either on a physical board or by means of any of the various chess-playing programmes. This paper describes a seminal initiative for chess schools; we propose a method for teaching and learning how to play chess, based on a web-based collaborative multimedia environment, called Chess Tutor. This collaborative environment facilitates the creation of groups in which one can learn the game by cooperating with other students and under the supervision of a Chess Master. We could point out quite a few remarkable aspects about this project, which has been running for about two years; however, this paper will focus on the cooperative scenarios used to carry out the teaching and learning process.

Keywords: chess teaching; collaborative multimedia environment; CSCW; computer-supported collaborative learning; CSCL; human–computer interaction; web application.

Reference to this paper should be made as follows: García, F.J., Moreno, M.N., García-Bermejo, J.R. and García, J. (2007) 'Collaborative multimedia environment for chess teaching', *Int. J. Computer Applications in Technology*, Vol. 29, No. 1, pp.54–62.

Biographical notes: Francisco J. García received a PhD in Computer Science (2000) from the University of Salamanca, Spain. He works as a Professor of the Computer Science Department at the University of Salamanca. He is the Director of the research GRoup in InterAction and eLearning (GRIAL). His main research interests are eLearning systems, web engineering, semantic web, human–computer interaction and software reuse.

María N. Moreno received a PhD in Chemistry (1998) from the University of Salamanca, Spain. She is a Professor of the Computer Science Department at the University of Salamanca. She is a member of the GRIAL research group. Her research interests include software metrics and data mining.

José R. García-Bermejo received a PhD in Physics (1989) from the University of Salamanca, Spain. He is a Professor of the Computer Science Department of the University of Salamanca. He has translated more than 30 books about computer science topics and is author of several books about object-oriented programming languages. He is interested in multiplatform programming, collaborative environments, visualisation and graphical user interfaces. He belongs to the GRIAL research group.

Joaquín García received a PhD in Education Sciences (1977) from the University of Salamanca, Spain. He is a full-time Professor in the Theory and History of Education Department at the University of Salamanca. He is the Director of the Laboratory of Multimedia Educational Design and Tele-Education in the Educational Sciences Institute in this University. He is the Editor of the Teoría de la Educación journal in its both editions (digital and physical). His research interests are related to the virtual educational spaces. He has published more than 100 papers in the education area. He is a member of the GRIAL research group.

1 Introduction

Chess has drawn the attention of thousands since time immemorial. In fact, can be said to be one of the oldest games known to mankind, since there exists some indication dating from the sixth century about a game between armies that was played on a 64-hole board. It is said that this game was invented by a Brahman called Sissa Ben Dari (Brunet i Ballet, 1890).

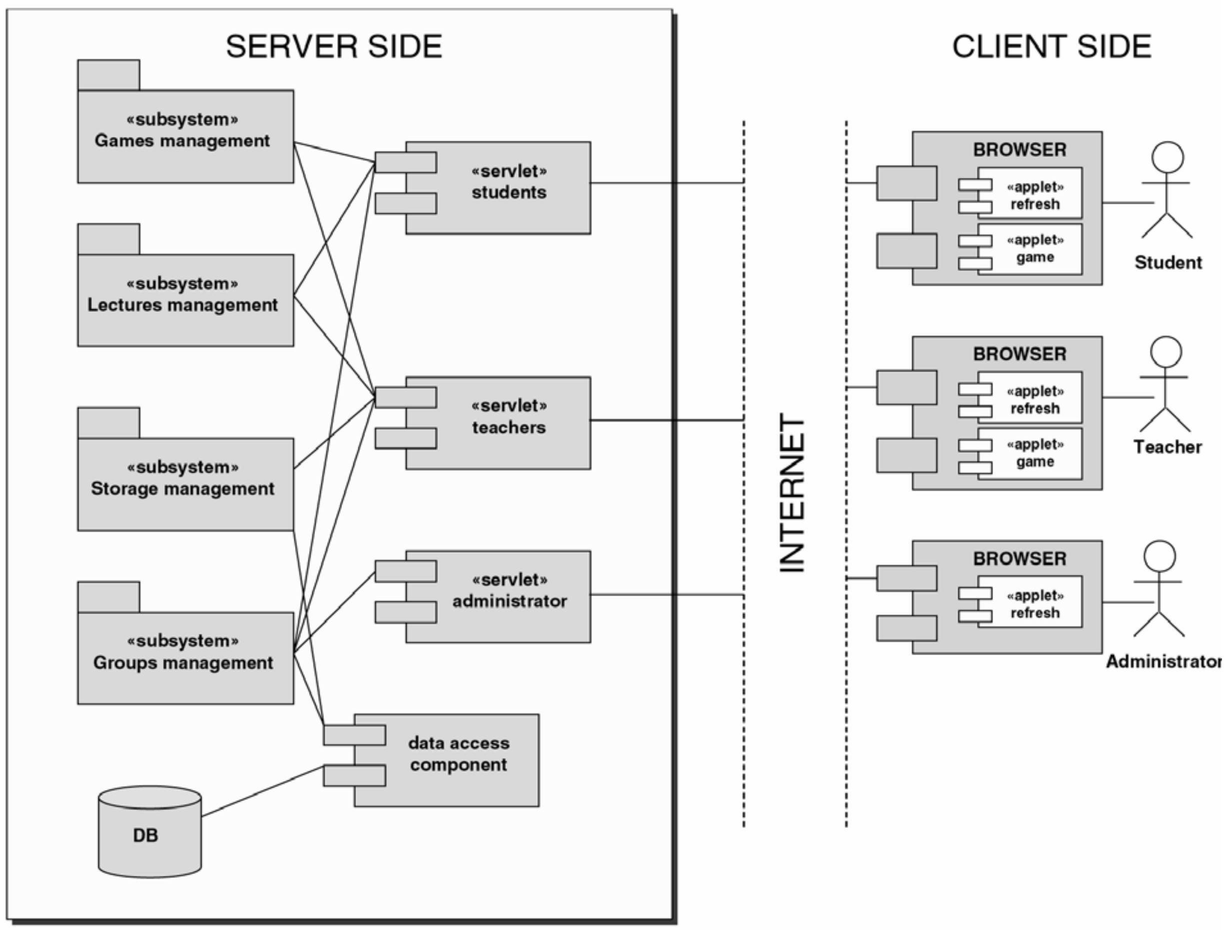
The link between chess and computers has existed since the very beginning of computing; it was the focus of much interest for the emerging community of artificial intelligence in the 1950s (Shannon, 1950; Turing, 1950). However, one must wait until 1958 for a first chess game, which was developed by Bernstein and Roberts (1958) that actually took into account all of the game's rules. Ten more years elapsed before one could find software good enough to be a worthy opponent in a match, this application was MACHACK VI, which was developed by a team, whose leader was Greenblatt et al. (1967).

Nowadays, the symbiosis between chess and computers is a fact, both in the research field and in the use of computers by those who play the game. Masters leverage the enormous potential of today's computers for their own training; also, at a more modes level, one can play interesting matches with 'virtual' adversaries or one can recreate and study mythical matches of the Great Masters (Shirov, 1997). Perhaps one of the most relevant facts in computer-chess history was Deep Blue (Campbell et al., 2002), the chess machine that defeated then-reigning World Chess Champion Garry Kasparov in a six-game match in 1997.

The effect of the internet, in this field as in so many others, has contributed to the spreading of chess, since it puts very useful tools in the hands of players, such as specialised portals and forums in which one can find news, programmes, motors for online gaming, etc.

However, all this potential for diffusion provided by the internet does not seem to have had a special repercussion in the teaching and learning process of this

Figure 1 Deployment scenario of Chess Tutor



game. Strangely, chess schools all over the world have incorporated internet to their pedagogic methodology by just making use of e-mail as an asynchronous means of communication between teacher and students, so that they can implement a tutorial action and exchange exercises.

Thus, most of the enormous possibilities of the web as a synchronous means of communication for this task are being wasted.

In order to fill this vacuum, a pioneer initiative appeared between *Escuela de Ajedrez Shirov On Line* (http://www.shirovonline.com) and the *Universidad de Salamanca*. Its purpose was the development of a collaborative multimedia web application that would focus on the pedagogic field, so that it could facilitate the teaching of chess through internet. It attempted to empower the interaction between a Chess Master and a group of students, but also between the students themselves. This was to be done by means of a set of collaborative scenarios, which must reproduce, as much as possible, the actual (real) scenarios by means of which the teaching and learning of chess takes place in a traditional Chess School. The fruit of this cooperation is the web application called Chess Tutor (*Ajedrez Tutor*, originally in Spanish).

This paper will focus on the collaborative scenarios of Chess Tutor. We will describe the way in which the various intervening parties interact. Thus, the paper is organised as follows: on Section 2, we expose briefly the architecture of our application, in order to make it easier to understand how interaction happens in the collaborative teaching and learning process. Section 3 details the various support tools provided by this system in order to support communications between the collaborative scenarios. Section 4 describes the main collaborative scenarios of the application; Section 5 reviews those projects that have any relationship to the one we expose here; one must say that, at the moment of developing the project and when we were writing this paper, we knew of no other similar initiatives. Finally, Section 6 concludes the paper and shows our conclusions.

2 Software architecture

Chess Tutor's architecture is that of a classic n-layer, thus following the deployment pattern of a web server with local business logic (Microsoft, 2002). Consequently, all of the application's components are on the server side. Figure 1 shows the main deployment scenario of the Chess Tutor architecture.

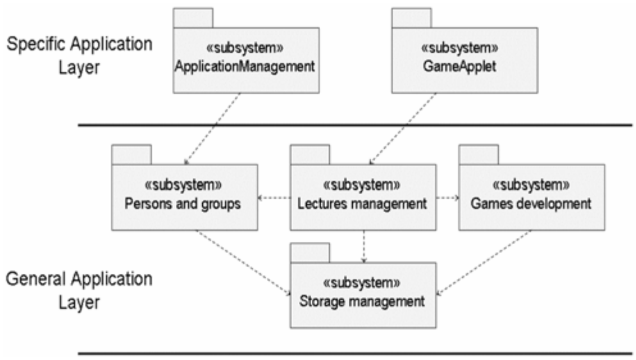
Hence, the server contains the data-access logic, besides data themselves, as well as the business and presentation logic. These layers are implemented as servlets (Armstrong et al., 2005). The most important servlets are the ones that implement the presentation layer, since they connect the user interaction layer with the application's business logic. They constitute the access point for the various users of the system: students, teachers and administrators. The functional subsystems implement the business logic and make it possible to managers who use them to develop different kinds of classes.

Some of the functional subsystems need to store data. There exist two types of data sources. One of them is the information about games, which is stored in files with Portable Game Notation (PGN) format (Edwards, 1994). On the other hand, there is the user information, for which we use a relational database, MySQL (http://www.mysql.com).

The clients, which are Java-enabled browsers, are not exactly light clients, since they must execute applets in which the interaction layer resides. To be precise, we have two applets (Sun Microsystems, 2005), *Game* and *Refresh*.

Figure 2 expresses the two upper levels of the logic architecture of the system. The specific application layer contains two packages representing interfaces to access to the global services of the Chess Tutor. The first one is the ApplicationManagement subsystem that is devoted to all set-up and general management issues of the application and the other one the GameApplet subsystem, which is implemented as an applet devoted to the development of the lectures and the games. The general application layer is formed by five main packages related to the management of the persons and groups, the development of the lectures and the games and the communication with the storage layer.

Figure 2 Upper layers of the software architecture

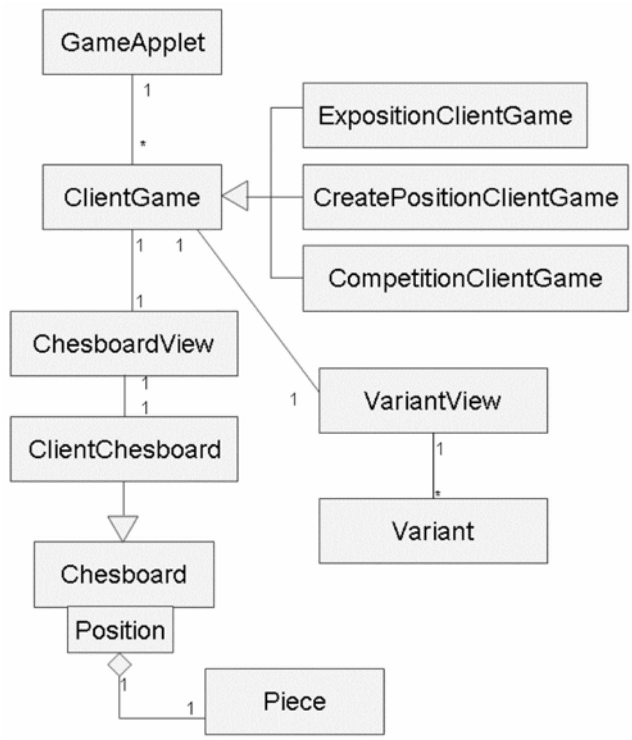


The Game applet has as its main purpose that of freeing the server of all tasks related with the management and placement of pieces on the board or boards. Figure 3 presents the main classes that support this applet.

The Refresh plays a most important role, since it implements the refresh mechanism by means of which we synchronise the clients with the server business logic. This process is essential if one is to get response times that are reasonable for the application's usability. Since browsers have a user-demand page download policy, it is necessary to have a page-demand mechanism that is server-controlled. In this way, in order to synchronise all clients, the Refresh applet acts like an agent in the client side that waits for the server's signal to refresh the board of each client; thus, the user can go on with the class with no delays.

Figure 4 shows an interaction diagram in which a Refresh example induced by the server (the start of a lecture) is explained. On the left side of this diagram the teacher starts a specific kind of lecture. This action arrives from the Teacher actor to the Group object (this one has the logic about the lecture start) through the teacher's navigator and Teacher servlet. Once the lecture is started, the student's pages should be refreshed (the teacher page must be refreshed too, but it is not shown in the diagram by simplicity). In order to refresh the student's navigator, the Group object uses the Refresh object, which is the object in charge of maintaining a stable connection with the client' applets. When the Refresh object receives the Group object request, it communicates it to the applets selected by the Group object. The communication between the Refresh object and client applet is asynchronous and it is made through a socket connection.

Figure 3 Class diagram of the Game applet



3 Communication support tools

Before describing the various collaborative scenarios of our application, we will first offer an overview of the tools used by Chess Tutor in order to facilitate communication and interaction as classes take place.

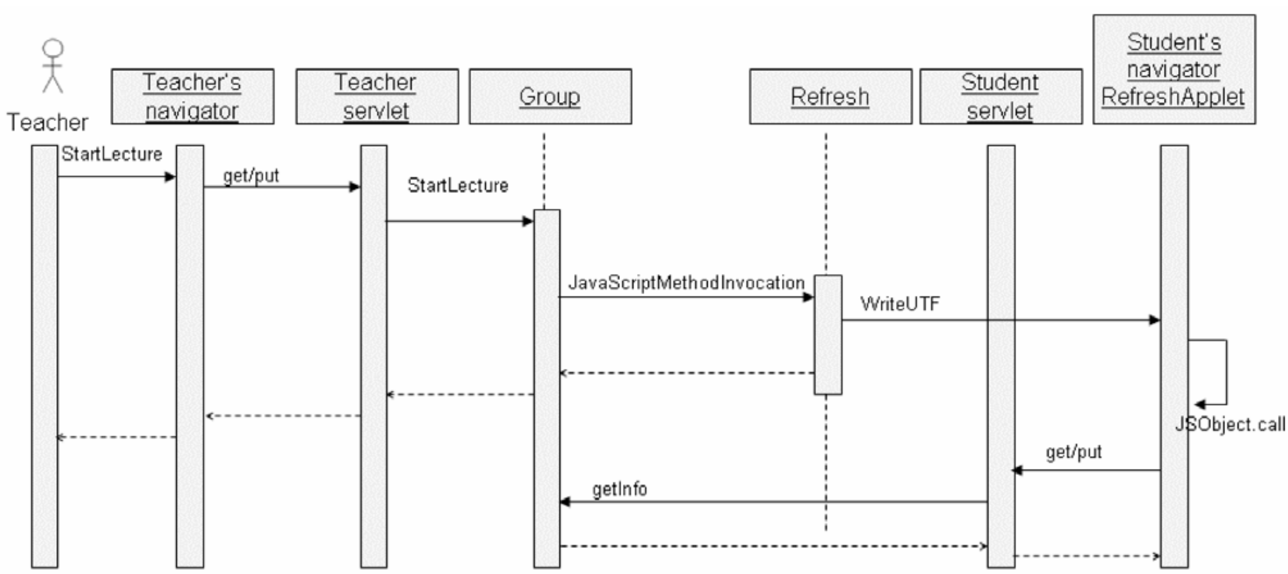
Even though it is not a communication tool, one must describe firstly the administrative side of Chess Tutor. This aspect of the program allows one to register both teachers and students and is used to set up class groups. Once groups have been established, students log in onto the Chess School with their own username, thus joining the class for their appointed group.

Taking a class is a synchronous activity that requires tools capable of allowing a fluid communication between teachers and students. This must happen independently of the scenario chosen by the teacher for a work session.

Chess Tutor provides an internal chat tool that works in all scenarios, thus allowing the group's components to communicate in writing. However, it was seen from the very first evaluations, which were carried out using early prototypes in extra courses at University of Salamanca on 2002/2003, that a simple text channel was just not enough for a fluid work session. Hence, it was decided to include a voice option and to use the chat as an auxiliary tool. For this purpose we have used external auxiliary tools, which are not built into the Chess Tutor application. Then, these applications should set up independently of the chess software looking for the best performance in lectures development, trying to reach a balance with all the variables that appear in a collaborative scenario (number of the student, level/age/maturity of the student…).

The first tool we used was *Paltalk* (http://www.paltalk.com/PalTalkSite/), a freely distributable application that allows full-duplex, one-to-many

Figure 4 Refresh scenario example



communication. This makes it ideal for collaborative scenarios, since it allows the teacher to talk to students and it also makes it possible for the student to talk to the whole group (both the other students and the teacher).

Paltalk is widely by students who take the course from their homes; this is also true when a Master is teaching an elite student. However, when students share a computer lab to follow classes, which is rather frequent in primary schools, this program is less than useful. Two problems come up; the first one is related to program installation and the interface language (which is English). A second problem is a loss of viability due to the fact that all of the students want to talk at the same time. This is complicated further by the noisy environment one gets when children between 7 and 12 years of age share a room.

Given the situation and not wanting to loose the capability of communicating via voice, we looked for an alternative based on radio streaming through internet. Thus, the teacher's computer turns into a radio transmitter and students listen to him or her by means of a receiver in their machines. There are several options for radio servers, both commercial and open source. As an example of free solutions one could mention *Shout Cast DNAS* (http://www.shoutcast.com/) or *Icecast* (http://www.icecast.org/), this last being the one we use nowadays. As a receiver, the best option is *WinAmp 5* (http://www.winamp.com/), which also offers free versions.

4 Collaborative scenarios

Chess Tutor provides five collaborative scenarios for teaching and learning how to play chess. They are as follows:

1 exposition
2 create position
3 competition
4 simultaneous matches and
5 exercises.

One must point out that *Simultaneous matches* and *Exercises* are but variants of the *Competition* scenario.

One of the key factors of this application concerning teaching and interaction is its ability to achieve simultaneity in classes. The possibility to carry out parallel classes about Competition, Simultaneous matches and Exercises offers great power and flexibility. This is specially true for large groups, since one can have several students competing among them (one to one), some other students competing individually against the teacher (Simultaneous matches) and the rest of them doing Exercises. All of this, of course, is done under the teacher's control and supervision.

4.1 Exposition

This scenario is the basic teaching unit, in which the teacher and a group of students share the same match. While the teacher has some privileges in the match, student interaction is under the teacher's control, by managing such permissions as *whites move*, *go back*, etc. This is similar to a contact class, in which the teacher explains and students participate while the teacher moderates.

The purpose of this scenario is purely pedagogic, although matches between students are both allowed and usual. However, from the point of view of interaction and from the perspective of the process of teaching and learning, the most important characteristic of this scenario is the treatment of variants. These are summarised in an area below the board (see Figures 5 and 6), in such a way that the teacher can decide, at any time, to create a variant of the game for study purposes. When a variant is created, it is added to the variant tree and the teacher can navigate existing branches in order to choose the one he or she thinks most convenient for each situation.

Figure 5 *Exposition* scenario. Teacher's view

Figure 6 *Exposition* scenario. Student's view

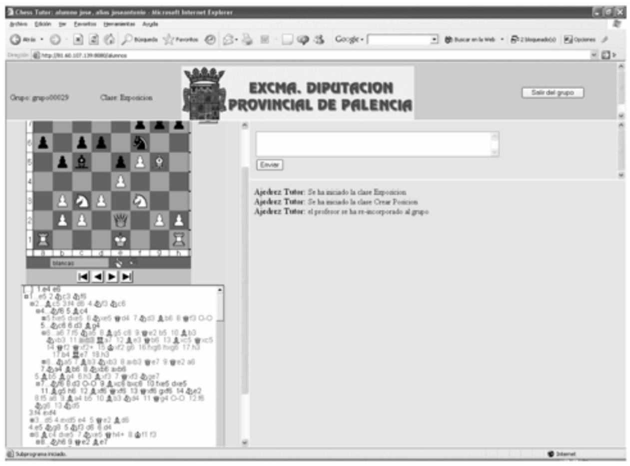


Figure 5 shows a view of scenario *Exposition*. We see a real situation, showing a group sponsored by *Excelentísima Diputación Provincial de Palencia (Spain)*. The match has been loaded from the match zone (lower left). When a student's id is marked red, the student is absent; when it is marked green it means a present student. On the lower right of the screen, student's permissions are shown. One can see that several alternative moves are compressed (when preceded by a + sign) and so they take up just one line; a – sign precedes unfolded moves. Both the teacher and the student enjoy a synchronised view of alternatives; that is to say, their view of the game's moves is the same.

The counterpart of Figure 5 can be seen in Figure 6. That is to say, Figure 6 shows the view for a student that belongs to the group of Figure 5. One can see how the structure of moves is the same both for the teacher's view and for the students' ones.

4.2 Create position

This scenario is essential to be able to study exits, endgames, special situations, etc. Besides, it must let the student propose situations so that they can be investigated and studied by the group.

The mechanics of usage is fairly simple. Pieces are placed on an empty board; this is done by whoever is indicated in the configuration. Normally, it will be the teacher and in this way the game will proceed as if it were an exposition. All students get an updated board so that they can follow the evolution of the match and they can participate either by chat or through *Paltalk*.

Figure 7 shows the mechanics of this scenario, in which one can see the classic endgame of bishop against knight, from the teacher's perspective.

Figure 7 Scenario *Create position*

4.3 Competition (Simultaneous matches)

If the *Exposition* scenario was the basic teaching unit, the Competition scenario (and hence the *Simultaneous matches* scenario) is the basic gaming unit, in which students compete against each other (Competition), or each student plays against the teacher (Simultaneous matches). By means of this kind of class, the tool can be similar to existing game programs, but it has a very distinctive characteristic: the collectivity produced by the group. In this way, the teacher must be able to control each game, whether between students or in a teacher-student game.

One of the most meaningful differences as compared to the *Exposition* scenario is the presences of a clock that measures each player's time, as well as the fact that several games are played at the same time and not just one.

The pedagogic principle that rules this scenario is the interaction between teacher and students or between the students themselves, with the mental process of the game as a background for that interaction.

One can see in Figure 8 the preparation done by the teacher before starting a Competition session; Figure 9 shows a view of the *Competition* scenario, in which it will be perceived that several of the clocks are actually in operation.

Figure 8 Preparing for scenario *Competition*

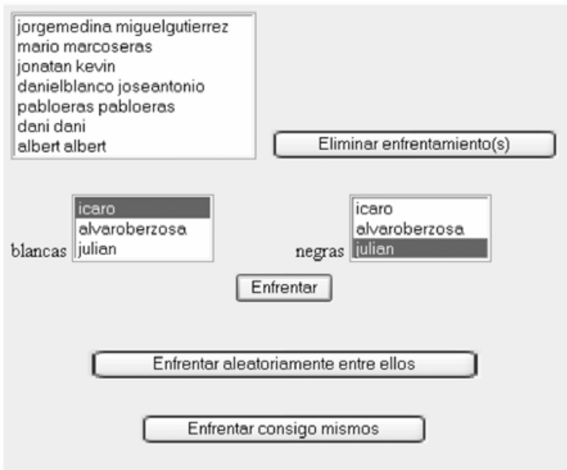
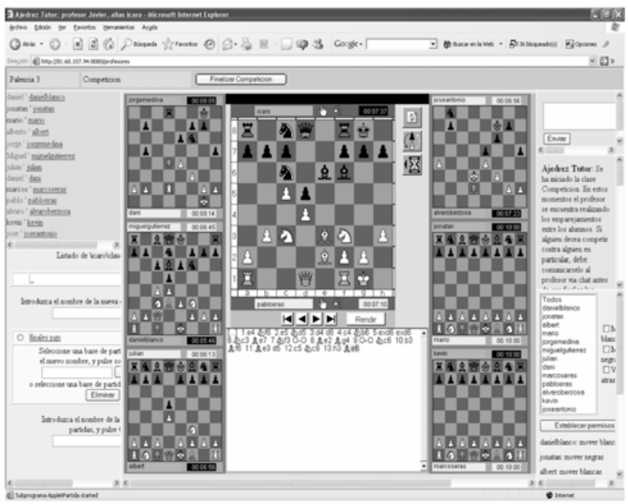


Figure 9 *Competition* scenario



It is also possible to load a game from the database, targeting several matches between students. For instance, Figure 10 shows the way to recover a game from the database and also the way to set up the competitions *miguelgutierrez-mario* and *dani-alvaroberzosa*. Besides, one can see the interface that makes it possible to select which games are to be saved and the possibility of setting some parameter for the PGN format.

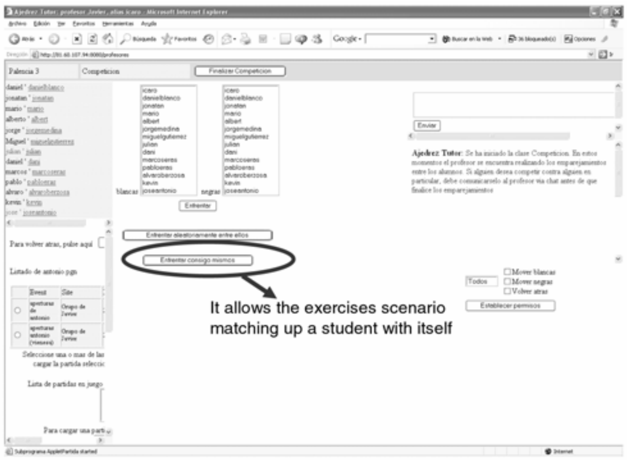
Figure 10 Recover/save games in *Competition* scenario

4.4 Exercises

This scenario represents another way of teaching. Its purpose is to be able to grade each student individually; this is done by means of the *Competition* scenario, playing a student against him or herself. We first create a base situation. The student, with his moves, adapts to one of the possible alternatives, thus getting a grade by means of a note or comment at the end of the exercise.

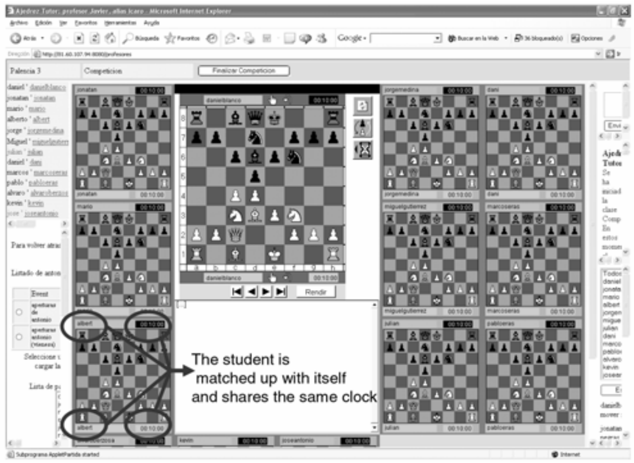
Figure 11 presents the confrontation screen. It represents an important tool for the management of the classes for the teacher. This tool allows to the teacher matched up two students for a game, using the match button (*Enfrentar* button in Spanish). Also the matching of the pairs of players can be made randomly, using the match randomly button (*Enfrentar aleatoriamente entre ellos* button in Spanish). But for the *Exercises* scenario this tool allows matching up one student with itself, using the match itself button (*Enfrentar consigo mismos* button in Spanish).

Figure 11 Matching up tool



In Figure 12 we can see the result of a previous matching up operation for exercises. It is like a Competition view, but with all the students match up with themselves and sharing the same clock, which is initialised to 10:00 min; this means that every student has 20 min to make the exercise, making the average between the black and white pieces.

Figure 12 *Exercises* scenario



5 Related works

Chess Tutor follows the conceptual framework that provides the Computer-Supported Collaborative Learning (CSCL) paradigm (Koschmann, 1996) because this collaborative environment allows the student learning chess approaching more complex situation and taking into account socially oriented learning theories that are based in the group strength.

There are many CSCL oriented systems such as DomoSim-TPC (Bravo et al., 2006) for domotical issues, COLER (Collaborative Learning environment for Entity-Relationship modelling) (Constantino-González and Suthers, 2001) for entity relationship modelling problems, C-CHENE (Baker and Lund, 1996) designed for teaching the concept of energy in physics, BetterBlether (Robertson et al., 1998) a computer mediated educational communication tool designed to facilitate and promote effective group interaction skills or Encouraging Positive Social Interaction while Learning ON-Line (EPSILON) (Soller and Lesgold, 2000) for object-oriented design.

In the field of chess we only have reference of a CSCL called ChessEdu (Mora and Moriyón, 2001b) that could be comparable with Chess Tutor. ChessEdu is a collaborative application that allows several people connected through a computer network participating simultaneously or asynchronously in a chess game. The main differences with Chess Tutor are that ChessEdu supports an asynchronous mode in order to analyse the history of game (Mora and Moriyón, 2001a), while ChessTutor is thought now for a synchronous learning, and also Chess Tutor is more flexible for learning because it presents more learning scenarios and always the class sessions are guided by a Chess Master, whereas in ChessEdu the tutor role is not compulsory. Besides, there are not documented experiences of the usage of ChessEdu by the chess community; however Chess Tutor is being used by several Chess Masters and International Masters to perform online courses and elite players training.

In the real cases of the chess schools, those which have some kind of internet presence do it by means of e-mail, o perhaps by offering a portal in which one can download games; once downloaded, games can be studied individually and independently by means of some of the programs with which a player can practice. This would be the case of *UNED* (http://www.uned.es/escuela-ajedrez/) or of the well-known *Escuela de Ajedrez Miguel Illesca*s (http://www.edami.com/).

The closest references are perhaps Internet Chess Club and Ajedrez21, although they are not dedicated purely to teaching. *Internet Chess Club* (http:// www.chessclub.com/), which was born on 1992 as an innovative idea in the world of chess. It is the internet's oldest chess service, which is why it suffers from legacy problems that stem for a command-line-oriented user interface. This program is executed locally and it connects to an internet server. Statistics show about 1000 players per day. It supports game databases. It is in communication with International Masters and Great Masters. It allows the broadcasting of international tournaments. It is oriented towards gaming and it allows competition between players. As the teaching of chess through the internet is concerned, its contribution is almost non-existent. *Ajedrez21* (http://www.ajedrez21.com/), to which one must connect by means of a Java-enabled browser, is the biggest chess server in the Spanish language (about 700 players per day). It sports an excellent user interface and is applet-based, as is Chess Tutor. Its main function is to offer matches between players; it provides several rooms and graphics for face-offs, depending on the amount of time and the player's level. Chat communication between players is allowed. However, support for game databases is schematic and game variants are not allowed.

6 Conclusions and further work

This paper introduces *Chess Tutor*, a web and collaborative multimedia environment based on the CSCL paradigm for chess teaching and learning that bets on a synchronous context for collaborative scenarios in order to achieve its purpose.

Its main goal is pedagogic, not that of facilitating meetings between players or the development of an engine by means of which one could set up challenges, which is what most products in the web do.

This application has approximately two years of experience, during which it has been the base for Chess School of one of the best chess players in the world, Alexei Shirov. It has been validated by a remarkable number of Great Masters and International Masters for training and courses; these are adequate for players with varying levels of ability (from elite players to novel ones) and of various ages (from primary school to graduate students). One can say then that this tool has achieved reasonable maturity; this can be seen in it interaction process for group collaboration with the aim of achieving the pedagogic goal sought. This is done by means of synchronisation mechanisms for the boards and both internal communication tools (chat) and external, voice-based open software tools.

Of course, this application must be improved. Firstly, the external communication tools are an easy solution in order to put our efforts in other areas of the learning process, however the integration of these tools and the development of new forms of communication are works to be done in a medium place. Secondly, if the exercises scenario is the weakest one and it needs more solid evaluation criteria, then it has to be redesigned. Not only the other scenarios will be revised in order to introduce usability improvements observed during these two years, but also we want to reinforce the tutoring action during the lectures.

Another weak point of this application is its only synchronous character. This means Chess Tutor is very good for groupware learning and living activities. This was our first goal and we do not want to lose this identity, but the introduction of asynchronous possibilities (such as reviewing past lectures, making personal exercises, etc.) would surely give more flexibility to the pedagogic model and process.

In the pedagogic area, new experiences with children will be carried out during the next academic course. We want to demonstrate the influence of the chess in the right development of the intellectual skills of the children who play chess (abstract reasoning, analytical, synthetic and decision-making skills and so on).

Acknowledgements

We would like to thank the research GRoup in InterAction and eLearning (GRIAL) of the University of Salamanca as well as the members of Escuela de Ajedrez Shirov OnLine, for their support and contributions to the writing of this paper. In addition, we want to thank to the Education and Science Ministry of Spain, National Program in Technologies and Services for the Information Society, Ref.: TSI2005-00960.

References

Armstrong, E., et al. (2005) 'The j2ee 1.4 tutorial', Retrieved on 14 June 2005, from: http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html.

Baker, M.J. and Lund, K. (1996) 'Flexibly structuring the interaction in a CSCL environment', in P. Brna, A. Paiva and J. Self (Eds). *Proceedings of the Euroaied Conference*, Lisbon: Edições Colibri, pp.401–407.

Bernstein, A. and Roberts, M.D.V. (1958) 'Computer versus chess-player', *Scientific American*, Vol. 198, No. 6, pp.96–105.

Bravo, C., Redondo, M.Á., Ortega, M. and Verdejo, M.F. (2006) 'Collaborative environment for the learning of design: A model and a case study in domotics', *Computers and Education*, Vol. 46, No. 2, pp.152–173.

Brunet i Ballet, J. (1890) *El ajedrez: Investigaciones sobre su origen*, Barcelona: L'Avens.

Campbell, M., Hoane, A.J. and Hsu, F-H. (2002) 'Deep blue', *Artificial Intelligence*, Vol. 134, pp.57–83.

Constantino-González, M.Á. and Suthers, D.D. (2001) 'Coaching collaboration by comparing solutions and tracking participation', in P. Dillenbourg, A. Eurelings and K. Hakkarainen (Eds) *European Perspectives on Computer-Supported Collaborative Learning, Proceedings of the First European Conference on Computer-Supported Collaborative Learning*, pp.173–180.

Edwards, S.J. (1994) 'Portable game notation specification and implementation guide', Retrieved on 14 June 2005, from: http://www.very-best.de/pgn-spec.htm.

Greenblatt, R.D., Eastlake, D.E. and Crocker, S.D. (1967) 'The greenblatt chess program', *Proceedings of the Fall Joint Computer Conference*, Vol. 31, pp.801–810.

Koschmann, T. (Ed) (1996) *CSCL: Theory and Practice of an Emerging Paradigm*, London: Lawewnce Erlbaum Associates.

Microsoft (2002) 'Application architecture for.Net: Designing applications and services', Version 1.0: Microsoft.

Mora, M. and Moriyón, R. (2001a) 'Collaborative analysis and tutoring: The FACT framework', *Proceedings of the IEEE International Conference on Advanced Learning Techniques (ICALT'01): IEEE Computer Society*.

Mora, M. and Moriyón, R. (2001b) 'Guided collaborative chess tutoring through game history analysis', in M. Ortega and J. Bravo (Eds). *Computers and Education. Towards an Interconnected Society*, Dordrecht, The Netherlands: Kluwer Academic Publisher, pp.243–250.

Robertson, J., Good, J. and Pain, H. (1998) 'Betterblether: A computer based educational communication tool', *International Journal of Artificial Intelligence in Education*, Vol. 9, Nos. 3–4, pp.219–236.

Shannon, C.E. (1950) 'Programming a computer for playing chess', *Philosophical Magazine*, Vol. 41, No. 7, pp.256–275.

Shirov, A. (1997) *Fire on Board. Shirov's Best Games*, London: Cadogan Books.

Soller, A. and Lesgold, A. (2000) 'Knowledge acquisition for adaptive collaborative learning environments', *Proceedings of the AAAI Fall Symposium: Learning How to do Things*.

Sun Microsystems (2005) 'The java tutorial', Retrieved on 14 June 2005, from: http://java.sun.com/docs/books/tutorial/.

Turing, A.M. (1950) 'Computing machinery and intelligence', *Mind*, Vol. 59, pp.430–460.