A Collaborative Case Study System For Distance Learning

Marta C. Rosatelli, John Self

International Journal of Artificial Intelligence in Education 14 (2004) 1-29
IOS Press

A Collaborative Case Study System For Distance Learning

Marta C. Rosatelli, *Programa de Pós Graduação em Informática, Universidade Católica de Santos, R. Dr. Carvalho de Mendonça, 114, Vila Mathias, Santos-SP, Brazil, 11070-906*
rosatelli@unisantos.edu.br
http://www.unisantos.br/mestrado/informatica/professores/marta/index.php

John A. Self *Computer Based Learning Unit, University of Leeds, Leeds LS2 9JT, UK*
jas@cbl.leeds.ac.uk
http://www.cbl.leeds.ac.uk/~jas/home-page.html

Abstract. Distance Learning from Case Studies involves enabling collaboration between two or more learners at a distance on a case study activity. In this paper we present an empirical qualitative study that simulates a learning scenario in which a pair of subjects at a distance are provided with a collaborative learning environment and required to collaborate in order to solve a case study. The results of this empirical qualitative study have implications that informed the design of a system: LeCS (Learning from Case Studies). LeCS is a web-based collaborative case study system that can be applied to any domain in which the learning from case studies method is used. It provides a set of tools for geographically dispersed students working collaboratively on the case study solution. In addition, it accomplishes functions regarding the learning process that together give real-time support to the distant learners during the development of the case study solution. The paper gives a general description of LeCS: its user interface, implementation, and agent-based architecture. Moreover, it describes how LeCS addresses the recommendations derived from the empirical study.

INTRODUCTION

As a result of the possibilities allowed by the new technologies we observe a rapid growth in computer-based applications for distance learning. Most of the systems developed focus either on providing tools for the teachers to design distance courses (e.g., WebCT, Lotus LearningSpace) or for distance course management (e.g., Preston & Shackelford, 1998). However, we believe that providing real-time support to the distant learners might be of greater relevance from an educational point of view.

In Rosatelli & Self (1999) we proposed a model for web-based distance learning from case studies that allows a group of learners to work on a case study at a distance. This model consists of an application of the case method to distance learning that makes uses of Intelligent Tutoring Systems (ITS) elements (Rosatelli, Self, & Thiry, 2000).

Learning from Case Studies is well established as an educational method (Christensen & Hansen, 1987). It has been widely used for years in disciplines such as law, psychology, psychiatry, architecture, education, engineering, business, and management. The common characteristic among those disciplines is that they introduce the kind of problems that no analytical technique or approach is suitable to solve: open-ended, with no “correct” or clear-cut solution. Hence, the case method is used where the skills of solving complex and unstructured

1560-4292/03/$8.00

problems are required (Easton, 1982). In addition, the method is considered a way to bridge the gap between theory and practice, creating opportunities for the learner to face the complexity of real problems and to deal with the day-to-day ambiguities of professional life (Shulman, 1992).

The case method is potentially appropriate to be applied to distance learning provided that fundamental issues in the design of distance education (Burt, 1997) are addressed. Rich (1995) used the web as a distribution medium for the case study contents, and a mailing list and e-mail to carry out the case discussion. But the web provides much more scope for support than this. As Oram (1996) suggests, it is also essential to assist the different stages of the learning process. The case study activity might be supported over the web with: (1) the use of text based and graphical hypermedia resources to present a case study and to lead the learner through the system use; (2) the use of tools to carry out discussions and group activities; and (3) the combination of both on-line and off-line case study activities, as they usually take longer than other learning activities. The characteristics of case study activities have led to their relative neglect in AIED research. The projects which have sought to adapt this research for the web (e.g., Brusilovsky, Schwarz, & Weber, 1996) have typically focussed on problem solving monitoring, rather than on case study activities.

Research on ITS has much to contribute to distance learning applications. Recently the distance learning area has experienced an increasing demand for the development of applications that make use of the new information and communication technologies (e.g., the web) as the medium for delivering courses. Intelligent distance learning can supply this demand and also provide real-time support to the distance learning process through the incorporation of ITS techniques. Thus, the work presented in this paper, besides introducing the case study element that is particularly novel in ITS research, shares characteristics with other approaches used in the area: agent-based ITSs (e.g., Shaw, Ganeshan, Johnson, & Millar, 1999), work on collaboration (e.g., Mühlenbrock, Tewissen, & Hoppe, 1998), and work on supporting the problem solving process at a distance (e.g., Greer, McCalla, Cooke, Collins, Kumar, Bishop, & Vassileva, 1998).

Moreover, the system described in this paper has features that are quite similar to the ones encountered in other collaborative and intelligent distance learning systems. In particular, COLER (González & Suthers, 2000), which is a web-based collaborative learning environment in the domain of database design, also focuses on both individual and collaborative learning, and on an agreement by the group on a joint solution for a collaborative task; monitors the students' participation and encourages the students to discuss their differences; and generates advice to the students concerning various issues.

The context for our research is a distance course on production engineering that includes disciplines containing open-ended problems about design, analysis, selection, planning, and/or business decision situations. Usually, such problems are derived from actual experience, reflecting the "real world" concerns of engineers and managers, and are used to train learners for professional practice. However, the system presented in this paper can be applied to any domain in which the learning from case studies method is used.

The paper is organized as follows. In the next section we give an overview of the Learning from Case Studies method, focussing on its most important features and the significance of such features to an intelligent system to support the case method. In the following section, we describe the empirical qualitative study, which simulated a learning scenario in which a pair of subjects at a distance are provided with a collaborative learning environment and required to collaborate in order to solve a case study. In the section thereafter we present an analysis of the empirical study results and their implications. Afterwards, we introduce LeCS (Learning from Case Studies) - a collaborative case study system that was developed based on the recommendations derived from the empirical study. This section gives a general description of the system: its user interface, implementation, and agent-based architecture. Next, we detail how LeCS addresses our recommendations. In the section that follows we discuss some relevant issues. Finally, we present the conclusions and directions for future work.

LEARNING FROM CASE STUDIES

A case study can take diverse forms such as a story, an event, or a text. It is an "instance of a larger theoretical class" and as such it represents certain features of that class. In addition to the narrative, cases are situated or contextualised in application, place, time, and so on. Their specificity and localism mean that their use as teaching materials takes into account the situated nature of cognition in the learning process. Also, case studies are suitable to present complex situations that demand cognitive flexibility to cope with them. As a result, the case method is often applied to learning in ill-structured domains (Shulman, 1992).

In the application of the case method in the traditional classroom, the case study will basically furnish raw material for the case discussion. This is a central issue in Learning from Case Studies and it is so important that the case method is often referred to as the process of teaching by holding discussions, as opposed to lectures or labs. The case discussion process is often described as fluid and collaborative. On the other hand, although it might seem at first to be freewheeling and unstructured, the discussion process has a kind of structure that usually emerges as it progresses (Hansen, 1987).

Easton (1982) provides a comprehensive framework for the case discussion, in which the case solution is developed step by step: the Seven Steps approach. The value of this approach to a computer-based system is that the case study solution is developed through a sequentially structured process, splitting it into parts that have a manageable grain size of information. The outcome of each step may be represented by a computer-based system so that it can interact with the learners, providing real-time support and feedback during the case solution process.

Each step of the Seven Steps approach has its own goal and suggests a range of activities to be carried out by the learners in order to achieve such goals. Table 1 presents an overview of the steps sequence, stating each step goal and exemplifying the kind of activities that are associated with each of them.

Table 1

The Seven Steps approach (adapted from Easton, 1982)

The Seven Steps	Activities
Step1. Understanding the situation	Relate, summarise
Step2. Diagnosing problem areas	List problems
Step3. Generating alternative solutions	List solutions
Step4. Predicting outcomes	List outcomes
Step5. Evaluating alternatives	List pros and cons
Step6. Rounding out the analysis	Detail, choose
Step7. Communicating the results	Present a solution to the case

In the Learning from Case Studies method the instructor role is intrinsically related with the case discussion. The leadership of the case discussion process is a critical responsibility of the instructor. Rather than having a substantive knowledge of the field or case problem he or she must lead the process by which individuals and the group explore the complexity of a case. The effective case instructor maximises the opportunities for learning by asking the appropriate questions during the discussion. The skills and techniques of a discussion leader are subtle but can be observed, abstracted, and taught. The attempt to identify question-and-response patterns typically leads to about eight basic types of questions (for instance, questions of fact, of interpretation, hypothetical questions, etc.) and dozens of subsets available for the case instructor's use (Christensen & Hansen, 1987).

THE EMPIRICAL STUDY

The application of the Learning from Case Studies method to a computer-based learning environment according to the Seven Steps approach that is proposed in our research takes into account how the case solution process occurs in the traditional classroom. The main issues to be considered are the guidance provided by the case instructor, the case discussion based on the confrontation of individual ideas, and the need to reach an agreement about the case solution. Below we describe the empirical study purpose, framework, and procedure.

Purpose

The main objective of the empirical qualitative study was to observe how the process of discussion between distant learners collaborating on the solution of a case study occurs. The need to carry out this kind of study was because although the discussion is a central issue in Learning from Case Studies in the traditional classroom, little or no information is available concerning the application of the case method to distance learning, using networked computers and a collaborative learning environment. Therefore, the empirical study aimed to further investigate the issues that arise from the case discussion in this kind of medium and environment. The experimental observations and results were then used to derive implications that inform the design of a system to support case study group activities in distance learning.

Framework

The framework on which the empirical study design was based included a set of factors that we consider influences student achievement. Concerning the learning environment, such factors are: the access to the case materials, the discussion medium, and the means of editing an answer. In addition, the actual pair of students, the time, and the case instructor role also have an impact on the student achievement. The empirical study design attempted to provide realistic conditions regarding the factors related to the learning environment

The collaborative learning environment that we set up for the empirical study was based on NCSA Habanero v2.0 Beta 2 (NCSA, 1998). Habanero is a collaborative framework written in Java that includes a client, a server, and a set of applications. Among the available set of applications in this version of the framework, the empirical study learning environment made use of (1) Savina, a collaborative Web browser; (2) Chat, a text based chat environment with logging capability; and (3) mpEDIT, a collaborative text editor (see Figure 1).

The use of these three particular tools was determined by the requirements imposed by the application of the case method combined with the Seven Steps approach in the traditional classroom. The collaborative web browser was a default assumption, as the empirical study tried to reproduce a distance learning scenario, making use of the web as the medium to deliver the teaching and learning materials. The collaborative aspect means that if one of the peers accesses a particular web page, the other participants in the same session would view the same page on their screens. Concerning the text editor, the collaborative feature allows that the participants can edit a text together, typing one at a time in the text area.

The web pages were the medium to deliver the case study materials. The first set of pages presented, besides the case study text, a brief explanation about the case method emphasising the importance of both the case discussion and the reaching of an agreement. They also included instructions to the learners on how to proceed to solve the case study according to the Seven Steps approach in this environment - that is, which tool to use, how, and when. The second set of web pages referred to the Seven Steps and its design was based on this approach. There was a web page corresponding to each of the steps. Each page included a description, the goal, the question, and the demanded case study activity concerning that particular step (see Table 1).

The questions associated with each of the Seven Steps were chosen from a set of examples modelled on approaches used by case study instructors in the traditional classroom (Meyers & Jones, 1993). Questions were elaborated according to a classification that takes into account the case instructor's purpose. The classification includes categories of questions such as: discussion starters, challenging questions, analytical and evaluative questions, summary questions and so on. In order to choose the appropriate questions for each step, we have matched these categories to the goal and requested activity of each step (see Table 1).

The hyperlinks of the web pages were designed to be followed sequentially, according to the nature of the approach: step-by-step. In each step the learners were supposed to answer individually the question posed. Then, based on their individual answers, they should have an on-line discussion in order to reach an agreement on a (group) joint answer. Proceeding through the sequence of steps conducted the learners to the development of a solution to the case study.

Collaboration (Roschelle & Teasley, 1995) turned out to be a main concern in the empirical study as the application of the case method demands it. The collaborative text editor was used to edit, to present, and to register both the individual and agreed group answers. The text based chat environment was where the discussion to reach an agreement - about both each step answer and the case solution - took place. The starting point for the discussion was the differences and/or similarities between the individual answers, that could be visualised with the collaborative text editor.

Although the context for our research is a distance course on Production Engineering, in the empirical study we used a case study on a general subject because the empirical study subjects were not attending a course or learning a particular domain, unlike the standard situation in the application of the case method. This particular case study did not demand from the participants who volunteered to take part in the empirical study any kind of specialised knowledge. Using a general knowledge case study eliminated any difficulty concerning the subject itself and therefore could provide more chance to observe issues related to the case discussion.

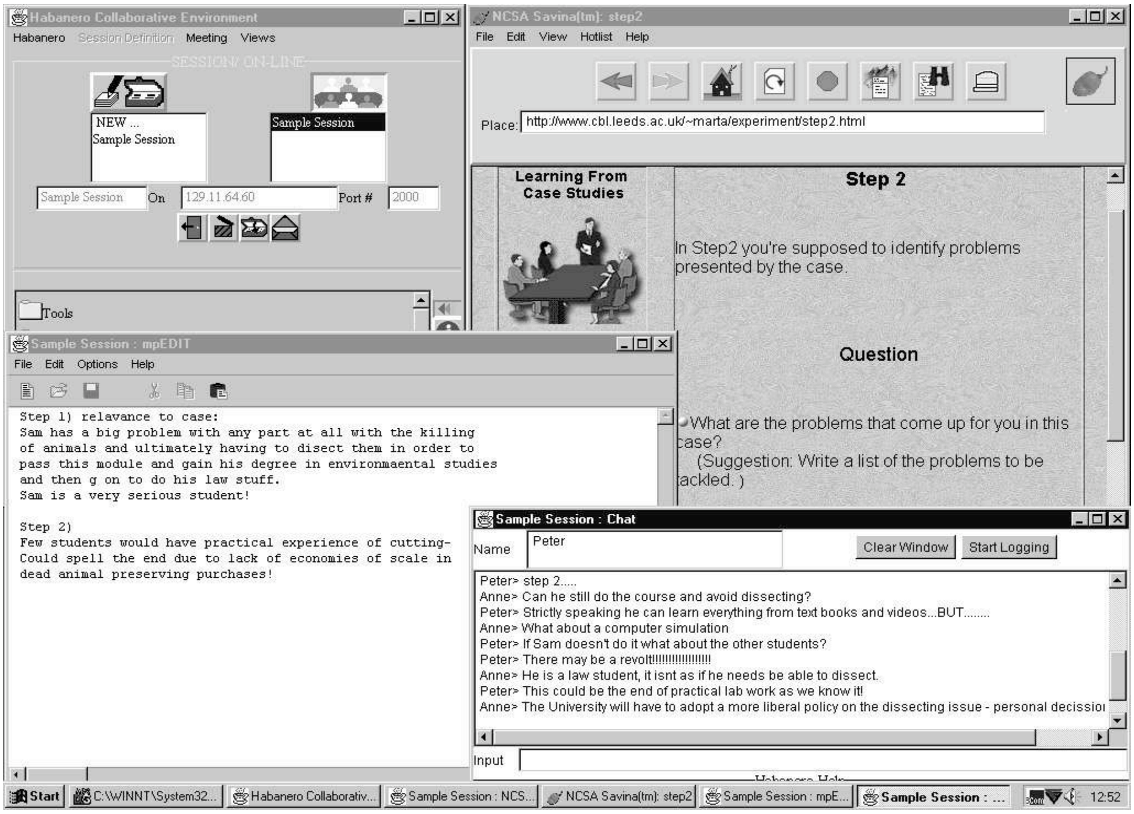

Fig. 1. Habanero Collaborative Environment screen showing on the upper left the framework main window, on the upper right the browser window, on the bottom left the text editor window, and on the bottom right the chat window

Furthermore, we believe that the controversial nature of the case study motivated and encouraged the discussion.

The case study used in the empirical study is entitled “A Dilemma Case on ‘Animal Rights’” (Herreid, 1996). In short, the case study poses a problem about what a professor in charge of a general biology course should do, when confronted with a student who is philosophically opposed to dissection labs. In this context, in order to solve the case study the subjects were supposed to consider the issues involved in the problem, examine the points of view and reasons of the “characters” (maybe even playing their roles), and get to a conclusion about what the professor should decide about the problem.

Procedure

The subjects for the empirical study were five pairs of postgraduate students, who volunteered to take part. Although they were from different backgrounds, they were familiar with the applications used in the empirical study: the browser, the text editor, and the chat. On the other hand, they were neither used to the collaborative aspect of the browser and the text editor, nor used to the case study method.

After being randomly paired, each pair of subjects received an explanation about how the empirical study would be carried out. This explanation aimed to briefly describe the kind of participation and interactions expected in the application of the case method. It emphasised the case discussion and the collaborative work on the case study solution process. It gave an overview of the Seven Steps approach and also stressed the reaching of an agreement, which was demanded in every intermediate decision point (that is, the agreed group answer to each step question) and in the final case solution.

The subjects were also given a 15-minute explanation about the Habanero Collaborative Framework. Two connected computers placed side by side were used to support this explanation. The objective was to demonstrate how the applications (the browser, the chat, and the text editor) operated under collaboration, and how to use them with the case method and the Seven Steps approach. Next, the subjects were located in separate rooms, using network connected computers (PCs under Windows NT) in order to carry out the empirical study.

One of the subjects of each pair was instructed to start the Habanero server and a Habanero client on his or her machine. The other subject started just a Habanero client, informing the IP number of his or her peer's machine. The peer in the machine that was running the server was instructed to save the chat log files and the edited answers text files. Afterwards, both subjects were told to fill out their names in the environment ID card and start working by accessing the web pages with the collaborative browser.

During the empirical study, the experimenter stayed part of the time with each of the subjects. Occasionally some assistance (only concerning the operation of the applications) was provided.

There were no constraints for the time the subjects would spend on solving the case study. They were just told that the empirical study was planned to take as a whole around 1 hour 30 minutes. As a result, after giving instructions, explanations, and setting up the environment, the estimated time to conclude the case study solution was supposed to be around 1 hour 15 minutes.

RESULTS ANALYSIS AND RECOMMENDATIONS

The main conclusion that we drew from the empirical study is that the application of the Learning from Case Studies method using the Seven Steps approach to a computer-based learning environment is feasible. We believe this is due to: (1) the web pages design, which was based on the Seven Steps approach, stating each step goal and posing a question; and (2) the collaborative learning environment, which provided appropriate tools for the different kinds of communication demanded by the case method combined with the steps approach. The empirical study generated the sort of dialogue that was expected concerning the case discussion, therefore validating our previous ideas about the development of the discussion process and the case study solution in a distance learning context. On the other hand, it also demonstrated that there are issues that should be addressed in order to provide a better support to group activity in Distance Leaning from Case Studies.

Concerning the results, all the five pairs of subjects were able to finish the empirical study, that is, they developed and presented a solution to the case study. According to the Learning from Case Studies method emphasis on the development of skills, we did not analyse the quality of such solutions. Furthermore, the empirical study purpose was not to evaluate how good the solutions were. Rather, we focused on how the case study solution was developed and how the participants responded to the problem posed by the case study, either thinking from the point of view of the case study characters or playing the role of these characters. Thus, the results analysis is derived from (1) the observations made by the experimenter; (2) the review of the log files containing the dialogue contributions; and (3) the review of the edited texts, concerning the individual and joint answers to each step question. In addition to this analysis and in order to follow the subjects' reasoning and solution path taken, the case solution was represented in detail in a tree data structure (Rosatelli & Self, 1998).

Below we detail our analysis of the empirical study results and the recommendations that inform the design of a system to support Distance Learning from Case Studies.

Solution Development

Although the subjects seemed to have a clear understanding of their task (that is, work collaboratively and present a solution to the case study through the accomplishment of the activities demanded in each of the Seven Steps), the experimenter observations and the log files review showed that they did not fully realise that the case solution is build up step-by-step, adding in each step to what they have done in the previous one.

Below we present the expected outcomes of each step according to the Seven Steps approach if the activities proposed in each of them are correctly followed:

- In the first step the subjects were asked to think about and list the relevant pieces of information presented by the case study. Thus, the outcome is a list of relevant information.
- In the second step they were asked to think about the problems that came up for them as they read the case study and write a list of the problems that they thought should be tackled. The outcome is a list of problems.
- In the third step they were supposed to list all the alternative solutions that they could foresee to the problems they have listed in the previous step. Thus, the outcome is a list of alternative solutions.
- In the fourth step they should say what could happen as a result of each action listed before, that is, to list the results of each alternative solution. The outcome is a list of the results prediction of each alternative solution.
- In the fifth step they were asked to list the pros and cons of each result (for the alternative solutions) they agreed on in the previous step. The outcome is a list of pros and cons for each result.
- In the sixth step they were supposed to choose among the alternative solutions they have listed previously. The outcome is a choice among the alternative solutions.
- In the seventh step they should present the answer to the question posed by the case study (the case solution developed). The outcome is the case study solution.

A consequence of such expected outcomes - particularly from Step 1 to Step 5 - is that managing the case solution development can become quite overwhelming and confusing, especially when it gets to the point of choosing a solution (Step 6). We observed that four out of the five subject pairs used some kind of identification for the components of the step answer. Two pairs numbered the step answer components for all steps; one pair either used numbers (Step 1) or items (Steps 2 and 3); and one pair numbered them for one of the steps (Step 2). This worked well for a single step. But, when moving to the next step, they often missed the interconnections between the answer components, therefore losing track of their solution paths.

This can be illustrated by the non-consideration of a certain topic raised in one step, in the subsequent one, demonstrating the abandonment of a possible solution path formerly envisioned[1]. For instance, from Step 3 (listing of alternative solutions) to Step 4 (listing of predicted results) the participants might miss one of the foreseen alternative solutions when listing the possible outcomes. Consequently one possible solution to the case study is not examined. Conversely, the opposite can happen: the inclusion of topics that were not anticipated in an earlier step.

In order to give support to this kind of problem the system should provide the learners with a way to map or visualise the building up of their solution. The development of the case study solution according to the Seven Steps might be well represented as a tree (Russell & Norvig, 1995). This kind of representation was proposed in (Rosatelli & Self, 1998) in which we, based on the empirical study data, derived a set of rules, developed and tried an algorithm to implement those rules, and generated the solution trees of the case study solution development for each pair of subjects. The solution tree representations show clearly the steps that were abandoned, the topics that were included in the middle of the process resulting from ideas that were not anticipated before, etc. Inputs for building up this tree are the outcomes of each step, that is, the solution tree is generated from the topics raised by the learners in each step answer. The root of the solution tree corresponds to the question posed by the case. The levels of the tree represent each of the Seven Steps. Each node in a given level refers to a component sentence of the answer to that step question. At a given step, one level is generated from the expansion of the nodes in the previous level and added to the tree. Thus, the solution tree corresponds to the representation of the case solution developed so far. Our recommendation is that, in order to point out any incoherence concerning the steps' expected outcomes, the system should generate and present a graphical representation of this solution tree to the learners during the case solution development.

Timing

On average, the subject pairs took 1 hour 26 minutes to complete the development of the case solution. We observed that they showed a tendency to spend more time in the earlier steps and go through the last ones quicker. However, this result was contrary to what was expected, as the last steps demand more complex activities. This can be due to various factors. The learners might (1) get overloaded with the level of detail demanded by the Seven Steps approach; (2) have a (normal) decrease in their productivity, so that they carry out better the earlier steps activities; (3) have just a rough idea of how the solution process will be developed according to the Seven Steps, concentrating too much effort in the first steps; and/or (4) take more time in the first steps in order to get familiar with the novel situation presented by the case study. Independent of the reason, the result was that when the subject pairs realised that they were taking a long time in responding to the earlier steps, they speeded up their work during the later steps, which might compromise the quality of their solutions.

To address this problem the system should have some kind of control of the time spent on carrying out the activities demanded in each step, particularly during on-line collaborative work.

[1] If the case solution is represented as a tree (Russell & Norvig, 1995), pruning is a valid technique. What is reported here is not pruning but not considering a branch of the tree just by negligence, without purpose or a further analysis.

This would be equivalent to the traditional classroom situation, where usually the instructor establishes a time limit for the accomplishment of tasks. The Seven Steps approach is particularly appropriate for a step-by-step time control. In a computer-based environment, the system should intervene in the learning process if the time spent by the learners discussing and answering a certain step question exceeds a certain time limit. This time limit should be a function of (1) an estimated time to answer each step question, and (2) an estimated time to solve a specific case study. In order to provide this kind of support, the system should perform the timing and warn the learners whenever they come close to the time limits.

Participation

The review of the empirical study dialogues and edited texts revealed a higher degree of participation concentrated in one of the peers, across the different phases of the case solution, in a significant number of situations. The partner in this case usually offered only a small number of contributions and an almost invariable acquiescence with his or her peer ideas.

In accordance with the case method theoretical foundations (cf. Learning from Case Studies section), the system does not aim to assess the learners' contributions individually. However, it should encourage the participation of all the group members. Therefore, if one of the learners remains silent or has a non-significant degree of participation, not contributing to the solution (for example, a small number of utterances mainly restricted to "I agree", "yes"), the system should be able to identify this behaviour and provide an additional stimulus to improve it, similar to the case instructor role in the traditional classroom.

Case-Specific Utterance Patterns

The empirical study dialogues revealed some relevant utterance patterns, that we classified as either discussion-specific or case-specific. The former - discussion-specific utterance patterns - referred to the process of negotiation to reach an agreement (for instance, "Do you agree?", "Can we move to step x?"). The latter - case-specific utterance patterns - were related to the step questions and to what was included in the previous answers, that is, related to the domain (for instance, "Sam does not like dissection labs."). Among these, we may focus on the case-specific utterance patterns that characterise a misunderstanding about the case solution, denoting difficulties in answering a particular step question.

The case-specific utterance patterns may represent points at which the system should provide support concerning the case study solution process, intervening in order to draw the learners' attention to a potential misunderstanding. In order to accomplish this function, there is no need that the expert knowledge be fully represented in the system. A case study may be represented in a script, a "standardised general episode" (Schank & Abelson, 1977). Such a kind of knowledge structure abstracts the sequence of events present in the case and can be used to support learners-system interactions (Parkes & Self, 1990). Similarly, the system should use the script-based representation to initiate interventions that are generated based on an identified misunderstanding about the case study and/or solution process.

Tools Coordination

Each tool used in the empirical study, namely the browser, the text editor, and the chat, served a different purpose, and were used at different points during the case solution development. Although the actions initiated by one of the subjects in a particular tool window were visible to his or her distant peer, the subject pairs often did not seem to be coordinated concerning switching between tools. This might be because each participant has his or her own pace to read, type, and reason.

This lack of coordination suggests that the capability of visualising the actions performed by the peer in a particular window is not enough to keep the group participants together, working as a group during the case solution development process. As a result, we believe that the system should be able to identify this lack of coordination and prompt a notification to the learners about their peer's actions. Every event detected on the screen of one of the learners (for example, a particular window is activated) should be checked against his or her peer's screen in order to verify if the other participants of the group have this same window active. If not, the learner should be notified that his or her peers are working on something else.

Synthesis of the Empirical Study Results

The empirical qualitative study results indicate that Distance Learning from Case Studies can be effective to the extent that the relevant issues in the application of the case method in the traditional classroom are taken into account and properly implemented in a computer-based system. That is, the pedagogy of the case study activity needs to be translated into a different medium.

In our proposal of implementing the case method in a distance learning scenario, the method's functioning principles are the premises for the system design and should be accomplished in two levels: (1) by providing the appropriate tools in order to make the case method operational in a networked environment, and (2) by adapting to this learning scenario the kind of support provided by the case instructor.

Regarding the tools to make the case method operational, as in the traditional classroom, guiding, promoting, and monitoring the case discussion is essential. In order to accomplish this we used the Seven Steps (Easton, 1982) to guide the case solution process. The step questions promote the case discussion, although this can be further improved according to the recommendations above. Concerning monitoring, the challenge is to use agent-based techniques within the customised tools of a collaborative framework to meet the recommendations derived from the empirical study.

With respect to the support provided by the case instructor, the main concern is the case discussion management. As the Seven Steps approach structures the group activity, it provides the system with a manageable grain size of information. As a result, the system should be able to handle two or more distant learners, taking account of the learning processes going on in each individual learner.

At this point it is important to clarify two aspects, concerning the application of the case method in the traditional classroom. First, usually the group size ranges from 3 to 4 students. The group size in a computer-based collaborative situation is similar. Second, the case instructor -

that has to manage 4 or 5 groups in the classroom - is often unable to take into consideration each individual participant of a group.

THE SYSTEM DEVELOPED

Bearing in mind what was discussed above, in this section we describe LeCS (Learning from Case Studies) - a collaborative case study system that we developed and implemented, informed by the recommendations derived from the empirical qualitative study. The design of LeCS is based on our model of supporting the learning from case studies method in a computer-based environment and in a distance learning context.

LeCS is a system for working collaboratively on a case study solution. It provides a set of tools and accomplishes some functions that together support the learners during the development of the case solution. The tools are a browser, a chat, a text editor, and a representational tool. The support LeCS provides consists of representing the solution path taken by the learners and making interventions concerning the following aspects of the case solution development: the time that the learners spend on each step of the Seven Steps approach (Easton, 1982); the learners' degree of participation in the case discussion; the misunderstandings that the learners might have about the case study; and the coordination of the group work.

LeCS was implemented in the Delphi language and has a client-server architecture. The server hosts sessions and the client interacts with sessions. A session is associated with a group of students working collaboratively on the solution of a case study. The clients run on the students' machines. The server can run on one of the student's machine or alternatively on a different machine.

User Interface

The LeCS user interface displays the following components shown in Figure 2: a *pull down menu,* and the following areas starting from the upper left, clockwise: a *participants' list,* a *browser,* a *solution graphical representation,* a *text editor*, a *chat* and a *system intervention area*.

The pull-down menu includes among others: (1) a *case studies library* containing the set of case studies that were modelled and are available to be worked on with the system; and (2) the *forms*, where the learners fill out the agreed group answer to each step question. There is a form to each step. The forms (Figure 3) are numbered and each entry corresponds to a component sentence of the respective step answer.

The participants' list shows all the group participants that are working on a case study. The participants who are on-line at a particular moment - logged on a certain session - have their names annotated with the green colour whereas the ones that are logged off are annotated in red. The participants can carry out tasks directly with another participant (send a message or see the information available about him or her) just by clicking on the button corresponding to his or her name (see Figure 2). Also, a timer is included in this area.

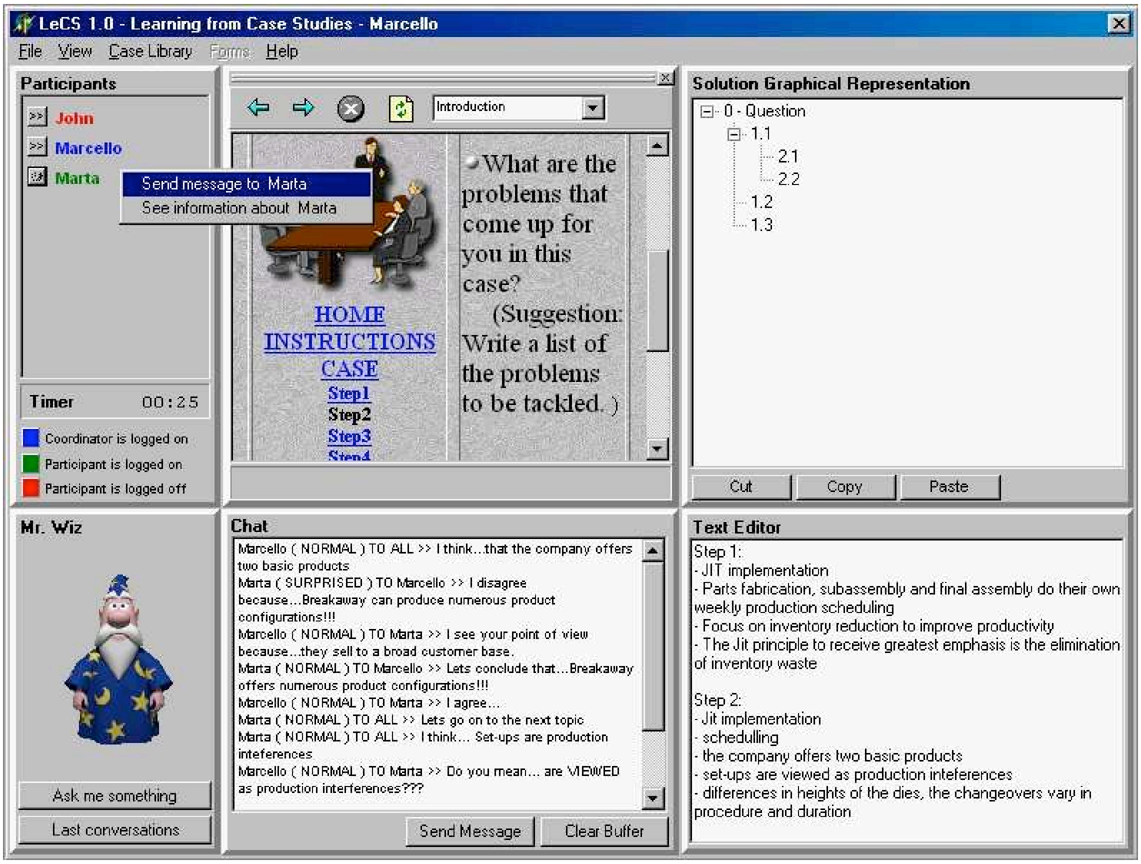

Fig. 2. The LeCS user interface

The browser is used to access the web pages that display the teaching and learning materials that guide the learners through the system use. The browser window is the only one that can be dragged and dropped and customised by the user.

The solution's graphical representation area displays the tree that is generated by the system during the case solution development. Thus, at each step that the group finishes, one level is added to the tree. The representation is displayed graphically as a directory tree. The nodes of the tree are numbered with the form's corresponding numbers. By clicking on the tree nodes - the numbers that correspond to the form items - the group can visualise the corresponding textual answer.

The text editor is an individual space where the learners can edit their individual answers. The text editor is used during the part of the solution process when individual learning takes place. The individual answers edited with this tool are supposed to be used when the learners participate in the case discussion (collaborative learning).

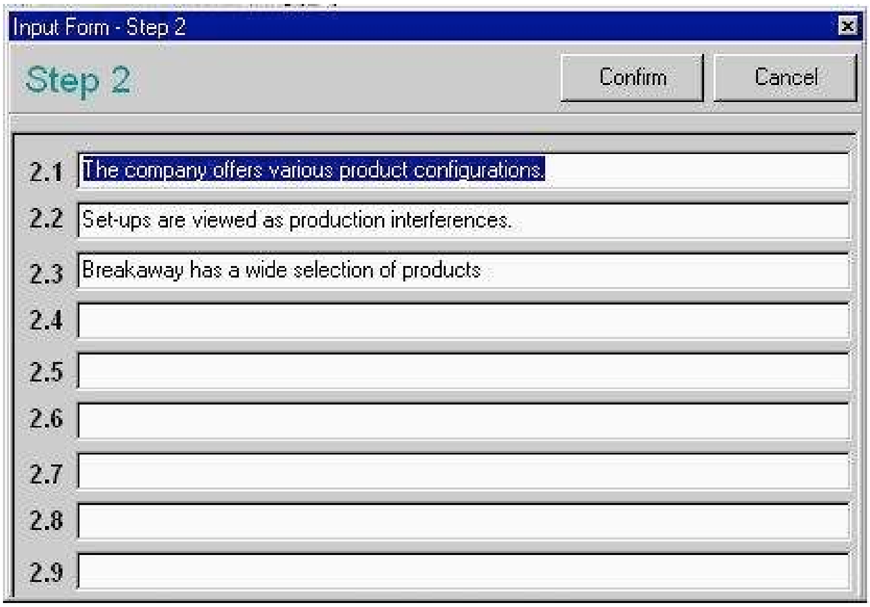

Fig. 3. The forms

The chat tool (Figure 4) is quite similar to the traditional programs of this kind and is where the case study discussion takes place. The participant can, besides writing free text, (1) express his or her emotional state; (2) direct his or her message to the whole group or to a particular participant (although this message is visible by all the group); and (3) make use of sentence openers (McManus & Aiken, 1995; Robertson, Good, & Pain, 1998) to facilitate the process of reaching an agreement in the case discussion.

The intervention area includes an interface agent that can be characterised as an animated pedagogical agent (Lester, Converse, Stone, Kahler, & Barlow, 1997). All the interventions that LeCS makes are presented through this agent.

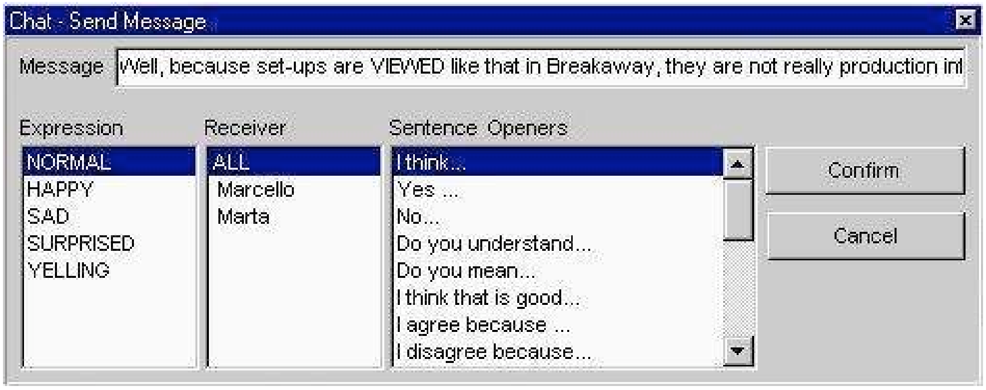

Fig. 4. The chat tool window

Architecture

The architecture description in this section outlines the agent-based (Franklin & Graesser, 1997) approach to tackle the issues raised in the empirical study result analysis and recommendations. The agents included in LeCS architecture can be classified either as interface, reactive, and/or hybrid agents (Nwana, 1996).

The LeCS agent-based architecture (Figure 5) is organised in a federated system. The agents' communication is based on an *Agent Communication Language* (Genesereth & Ketchpel, 1994). The messages exchanged between the agents use the KQML (Knowledge Query and Manipulation Language) format (Labrou & Finin, 1997). The communication structure establishes that the communication does not happen directly between agents, but rather through a *facilitator*. The facilitator is a special program - implemented as an agent - that keeps the information about each agent in the system, and is responsible for routing the messages, working as a broker. In addition, two databases were implemented: in the first one, the facilitator stores all the necessary information in order to route the messages; in the second one it logs all the exchanged messages.

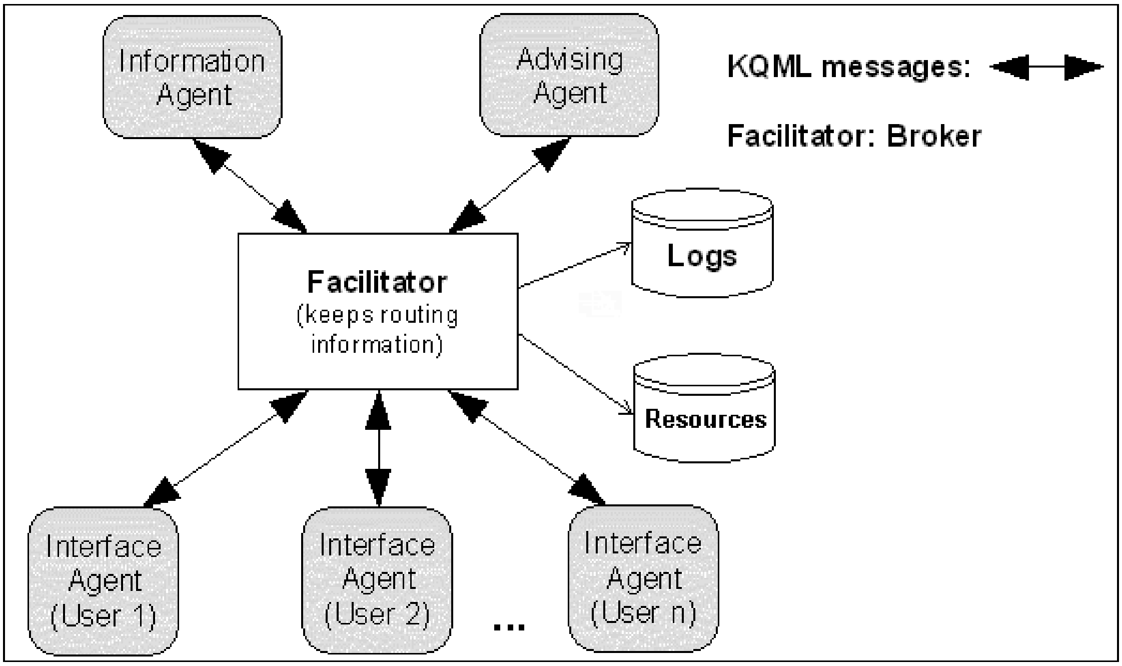

Fig. 5. The agent-based architecture

The LeCS architecture includes three classes of agents: interface agent, information agent, and advising agent. There is one information agent and one advising agent running during a session, but as many interface agents as there are participants logged on. In this context, the interface agents interact with the learners, provide information to the other agents, and initiates the system's interventions; the information agent deals with the domain and the pedagogical knowledge; and the advising agent monitors the learning process and also initiates system interventions.

Interface Agent

The interface agent (cf. Figure 2) can be characterised as an animated pedagogical agent (Lester *et al*., 1998). It resides on the participant machine and all the system interventions are presented through it. In addition, the interface agent stores information about the individual users: what is typed in the text editor, the number of contributions in the chat, the current step he or she is working on, the answer to each step question, and the time spent on each step (these last two are functions accomplished just by the interface agent of the group coordinator). Based on this information, the interface agent generates the interventions about timing and participation.

A resources database contains additional information: the agent address, the name by which it is known, and its network mapping. A history database is implemented to log everything the interface agent does, including the communications with the user and the other agents. The information agent and the advising agent also have these same kinds of databases.

Information Agent

This agent stores information that is divided into two different categories: didactic material and knowledge bases. Didactic material consists of HTML pages, images, and text. The knowledge bases refer to the domain and the pedagogical knowledge: the interventions about case-specific utterances (cf. Case-Specific Utterance Patterns section) and the representation of the case solution developed. The information agent also stores the chat interactions. Both the interface and the advising agents can access the information agent.

Advising Agent

While the information agent has knowledge base and database access features, the advising agent has engines to reason about the user's actions, and to recognise situations in which some support is needed. The advising agent executes an algorithm to generate the solution tree representation with the information provided by the interface agent, and returns this representation to this agent. With the information provided by the information agent, it identifies the case-specific utterance patterns that denote a learners' misunderstanding of the case study. When applicable, it generates and sends the request of an intervention about a misunderstanding to the interface agent. It also generates and requests to the interface agent interventions concerning the coordination of the group work and the jumping of a step.

ADDRESSING THE EMPIRICAL QUALITATIVE STUDY RECOMMENDATIONS

In this section we outline how the current version of LeCS addresses the recommendations derived from the empirical study results. We describe the means by which the system reasons and accomplishes the tasks involved in providing support to the learners. This, in a final analysis, determines both the sources of information for the agents (for instance, the knowledge bases) and how they perform their tasks (for instance, the generation of interventions).

Solution Development

According to our empirical study recommendations LeCS dynamically generates a knowledge representation, a tree data structure (Russell & Norvig, 1995) that represents the case solution development according to the Seven Steps approach (Easton, 1982). The tree represents the solution path taken by the group to solve the case study and is referred to as the solution tree. The solution tree is presented to the learners on the user interface. This means that what is displayed represents the case solution developed so far by the group (cf. Figure 2). The objective of this representation is to help the learners choose the best envisaged solution. All the group participants see the same representation. Furthermore, the graphical elements in this window are editable by the learners when they disagree with the system reasoning, that is, with what is represented in the tree (for example, when the link made between a node and its parent node is not what the learners meant in their textual answer).

In order to generate this representation the connections between the tree nodes (the arcs) have to be obtained. To accomplish this the system has to make a link between a node in a given level to its parent node in the previous level through some kind of reasoning. A set of *if-then* rules (see Table 2 for a description of the rules) define our heuristics to make these connections (Rosatelli & Self, 1998). Such rules were derived from the empirical study and an algorithm implements those rules.

In this algorithm the tree nodes are labelled by the sentences and the arcs are labelled by the rules. Inputs are the outcomes of each step, namely the sentences that compose the learners' joint answers to each step question. The root of the tree corresponds to the question or dilemma posed by the case study. The levels of the tree represent each of the Seven Steps. Each node, in a given level, refers to a component sentence of the learners' joint answer to that step question. The algorithm has a procedure to compare the answers of two subsequent levels (steps). This comparison aims to find out which sentence in the previous level is related to the sentence that is being analysed in the current level.

The procedure to compare the sentences verifies if a set of keywords in a sentence of a given level is also included in a sentence of the previous level. This set of keywords is defined in every step when the group coordinator fills out the form with the group answer to the step question. Thus, the set of keywords of a level n sentence is compared with the words of a level $n-1$ sentence, word by word, until either one of the keywords is found, or the set of keywords is finished. In the former case (that is, if a keyword is found) the link between the nodes is made. The expansion of the nodes in a given level then is represented precisely when the nodes in that level are linked with the parent node in the previous level.

Next, provided that the interface agent informs the advising agent about which step the learners are currently working on, one level is added to the tree, and the procedure to compare the sentences is executed again. The final state of the tree represents all (ideally) alternative solutions that can be generated through the Seven Steps approach.

Example

Based on the review of the dialogue contributions and the edited texts from the empirical study we built solution trees for each of the five pairs of subjects (Rosatelli & Self, 1998). The aim was to test our heuristics and the functioning of the algorithm that we developed, as described above. The dialogue extracts were taken from the chat log files and text editor saved files[2].

Table 3 shows part of a transcript of the session of one of the subject pairs. For a better understanding, the parts of the dialogue that referred to the joint step answers were selected, edited, and conveniently enumerated. That is, we extracted from the text the step answers and changed or numbered them according to the nodes in the solution tree graphical representation.

Figure 6 displays the solution tree generated for this subjects pair. The solid line indicates the links made using the if-then rules, denoted by R2, R3, R4, R5, and/or R6. The long dash line indicates the cases when the nodes would be linked in a more correct way according to our understanding of the dialogues (that is, the system reasoning would be different from what we think is correct). The dash dot dot line are the links that students would supposedly do, when asked by the system. These cases are also identified by the use of R7.

Timing

Concerning timing, LeCS interventions consist of warning the learners when they exceed the time limits established for on-line collaborative work. An example of a timing intervention is "You are taking too long to complete this step". Such interventions are generated by the interface agent, which controls the time spent by the learners in each step. In accordance with our recommendations, the time limits should be a function of both the total time estimated for the solution of a given case study and the time to respond to the particular step they are working on. In LeCS the time limits per step for the case studies that are included in the library are set by the case instructor.

Participation

In LeCS the interface agent is also responsible for identifying and intervening regarding a low degree of participation of one or more learners in the case solution development. This is denoted by a small percentage of contributions of one of the peers during the case discussion. To accomplish this, the interface agent controls the learners' contributions in the chat per step. If a learner remains silent for more than 50% of the estimated time for working on a particular step the interface agent generates an intervention inviting the learner to participate. An example of a participation intervention is "Would you like to contribute to the discussion?". It is worth noticing that LeCS does not evaluate the learners' participation qualitatively.

---

[2] At first the extracts were supposed to be taken just from the text editor files but there were situations when the subjects answered the step questions directly during the chat.

Table 2

Set of *if-then* rules

R1
```
If level=0 then
    the question posed by the case study is the only node, the root
else
```

R2
```
   If level=1 then
       all the nodes in level 1 are linked to the root
   else
```

R3
```
      If level=2 or level=3 or level=4 or level=5 or level=6 or
      level=7 then
```

R4
```
         If number of sentences (nodes) in level n is equal to the
         number of sentences (nodes) in level n-1 then
             link nodes in level n to the correspondent one in level
             n-1
```

This rule is due to the fact that when the number of components sentences in one level is the same as the number of component sentences in the previous level usually there is an ordered correspondence between the learners' answers in two subsequent steps (levels).

R5
```
         If the sentence (node) in a given level n-1 contains the
         chosen keyword in level n sentence (node) and the node in
         level n is unlinked then
             the node in level n is linked to the node in level n-1
```

This rule implies that the link is made as soon as the chosen keyword is found in a level *n-1* sentence. The other sentences in level *n-1* are not examined. Also, this rule allows parent nodes in level *n-1* with only one link to nodes in level *n*.

R6
```
         If there is any unlinked node in level n and the sentence
         (node) in a given level n-1 contains the chosen keyword in
         level n sentence (node) then
             the node in level n is linked to the node in level n-1
```

The rule above accounts for a parent node in a level *n-1* with a second, a third, and so on, nodes in level *n* connected to it. A still unlinked node in level *n* is compared against an already connected node in level *n-1*. If the condition is satisfied, the nodes are linked. Note that the rule is basically the same as R5 applied to unlinked nodes in level *n* and not constrained to unlinked nodes in level *n-1*.

R7
```
         If there is any unlinked node in level n then
             ask the learners to link the unlinked node(s) in level n
             to its parent node in level n-1
```

This last rule is used as a latest resource, when it is not possible to connect one or more nodes in level *n* with a node in level n-1 using the previous rules. In this situation, the learners would be asked to make links themselves.

Table 3
Part of a transcript of a session of one of the empirical study subject pairs

Step 2
...
>Could Animal Model being used to replaced real animal in Biology Laboratory (2.1)
>Should the Biology student dissect the animal when they study about that, because they (another student) always do it? (2.2)
>How effective the Biology experiment without using real animal (2.3)
...
Step 3
...
>Making the models of animal that look like similar, like synthetic animal made from poly, plastic, or whatever (3.1)
>Divide the class in two session one use real animal second one use model or using multimedia or refer to video clip. (3.2)
...
Step 4
...
>Many manufacture or factory built (4.1)
>The number of population of animal which usually being used in the lab will be increased. (4.2)
>The student can't feel the "real" experiment (4.3)
>Lecturer will take longer time to conduct the experiment since they need to handle two classes (4.4)
>Conservation group will oppose the idea since animal dissection still happen. (4.5)
...

Case-Specific Utterance Patterns

The interventions that LeCS makes regarding case-specific utterances are based on what is modelled in the domain knowledge base concerning the case study. In order to represent this knowledge we have adapted the idea of constraint-based modelling (Mitrovic & Ohlsson, 1999). Constraint-based modelling (CBM) proposes that a domain be represented as constraints on correct solutions. As such it is appropriate to the identification of case-specific utterance patterns that characterise a learner misunderstanding the case study contents.

Based on the empirical study, we recommended (Rosatelli & Self, 1999) the use of scripts to support this kind of learners-system interaction. Similar to Parkes and Self (1990), the instructional aspects of the domain would be represented using the conceptual graphs notation (Sowa, 1984) and the system would have to "translate" the learners' utterances into the representational language and vice-versa. However, during the system development it became clear that scripts as the domain representation are appropriate and necessary only if the learners are to ask the system questions. CBM alone proved to be sufficient for LeCS to accomplish the function of intervening about the case-specific utterance patterns.

In CBM (Mitrovic & Ohlsson, 1999) the unit of knowledge is called a state constraint and is defined as an ordered pair <*Cr, Cs*>, where (1) *Cr* is the relevance condition that identifies the class of problem states for which the constraint is relevant, and (2) *Cs* is the satisfaction condition that identifies the class of relevant states in which the constraint is satisfied. For our purposes *Cr* is a pattern that indicates that a particular sentence is a case-specific utterance and *Cs* is the correct case-specific utterance. The semantics of a constraint is: if the properties *Cr* hold, then the properties *Cs* have to hold also (or else something is wrong). In LeCS, when "something is wrong", it means that there is a learner misunderstanding about this particular sentence. As a consequence, the system should initiate an intervention.

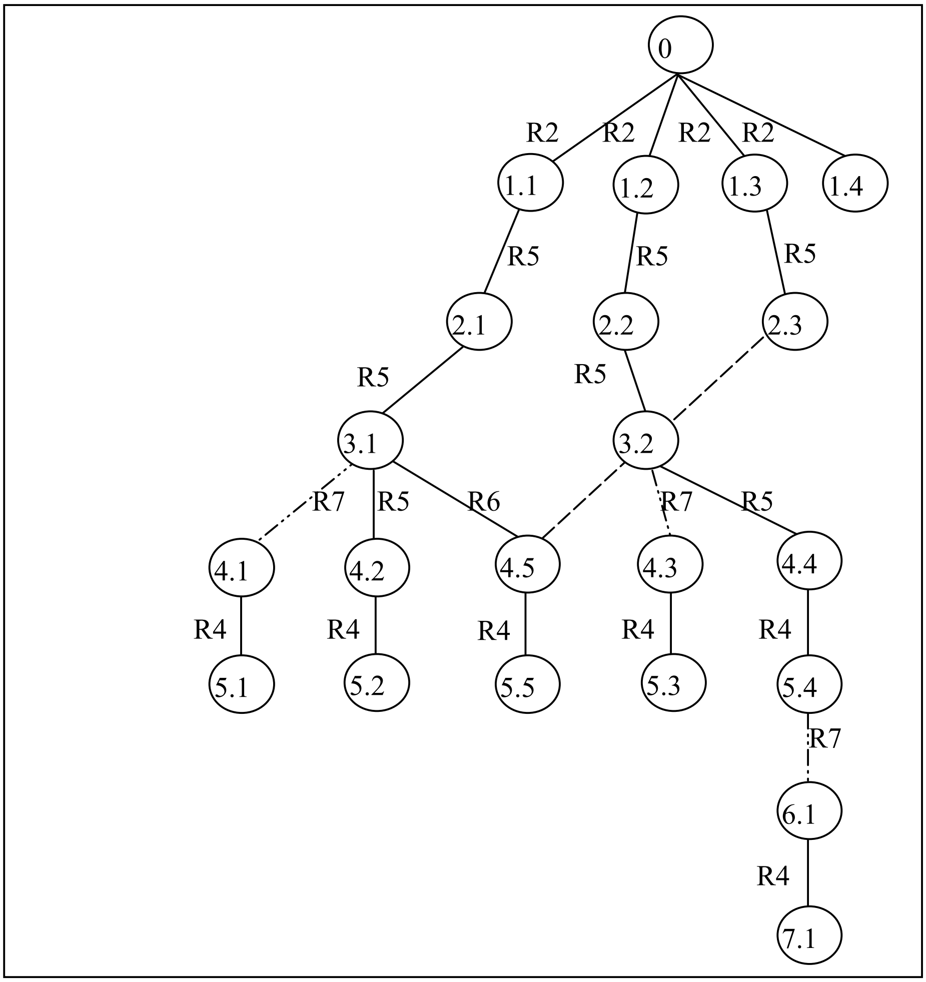

Fig. 6. Example of a solution tree

In order to initiate the intervention, LeCS checks for any violations of what is modelled. Hence, if the satisfaction pattern of a relevant constraint is not satisfied, then that state violates the constraint. In LeCS the problem states are the input sentences, that is, the sentences that compose the group answers to each step question, which are entered by the group in the forms. The intervention consists of stating the correct sentence.

Table 4 presents examples of constraints that refer to a case study in the Production Engineering domain, in the subject of Inventory Control and Materials Management (Tersine, 1988, p. 424). These constraints are modelled based on an analysis of this particular case study from the point of view of the case instructor regarding what kind of misunderstandings it could originate. In this case, an example of an intervention about a case-specific misunderstanding is “The company offers numerous product configurations”.

Table 4

An example of a state constraint (<Cr, Cs>) from a case study in the Production Engineering domain

If the “company offers” (Cr) then it has to be “numerous product configurations” (Cs).
If “Breakaway is recognised by” (Cr) then it has to be “its wide selection” (Cs).
If “set-ups are” (Cr) then it has to be “viewed as production interferences” (Cs).

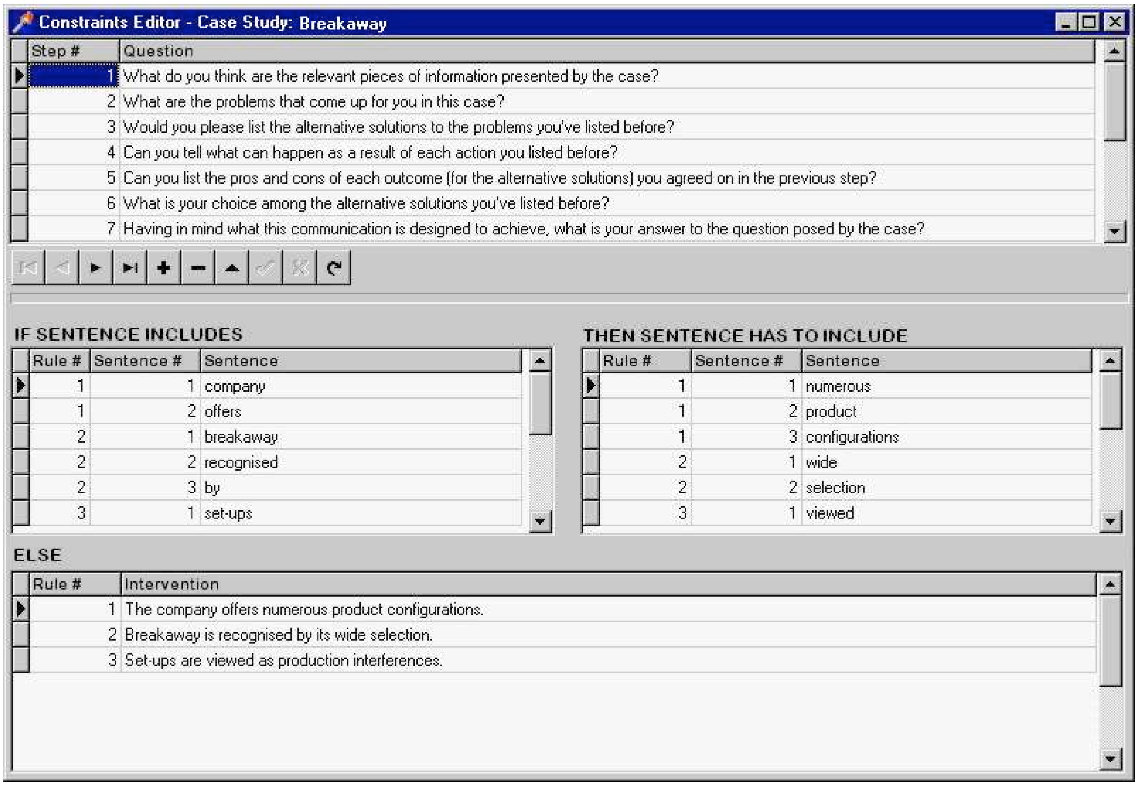

Fig. 7. The LeCS user interface for the case instructor modelling a case study

Figure 7 shows the LeCS user interface for the case instructor modelling a case study. The system interventions concerning case-specific utterance patterns are generated based on the constraints edited by the case instructor using the forms provided by this user interface. It is important to note that, in the current implementation, the order in which the words are used by the learners does not make any difference to the system response. Also, the constraints editor allows the case instructor to edit as many constraints as he or she deems necessary or appropriate for a given case study.

Tools Coordination

According to our recommendations, the system should prompt a notification to the learners when a lack of coordination in using the tools is identified. However, in the current implementation of LeCS we do not have this level of control upon the learners' actions. The system only coordinates the group through the Seven Steps. In order to accomplish this, when starting a session the group should define a coordinator: a member of the group who is responsible for filling out the forms with the group's joint answers to the step questions. In this way the contributions gleaned from the various participants regarding the step answers are integrated into the joint answers. This means that the forms are disabled to all the other members of the group except the coordinator and the coordination function is accomplished by LeCS based on the group coordinator actions. He or she, in a certain sense, defines to the system the pace at which the group proceeds in the solution development. Thus, once the coordinator fills out the forms of a particular step question and then moves on to the next step, the system "knows" which step the group should be working on. If any of the learners, for instance, accesses a web page different from the one referring to that step, the system will prompt him or her a notification, warning that the group is working on a different step. An example of an intervention about the tools coordination is "The group is working on step x". At any time during the solution process the group can decide to change its coordinator.

The higher degree of coordination that we recommended based on the empirical study is not necessary in LeCS as: (1) the text editor is individual and not shared as in the empirical study that we carried out, and (2) the group joint answers are typed in forms rather than in the text editor. Therefore, the learners use the chat to communicate and make the joint decisions. Besides, they do not have much opportunity to interfere with each other's actions in editing the individual or group answers.

Additional Issues

Besides the recommendations that we made based on our empirical study, there are some additional issues that came into view during the LeCS implementation. Below we describe such issues.

Jumping a Step

In the empirical study, three out of the five subject pairs jumped a step during the case solution development. In LeCS we do not constrain the learners by requiring them to complete all the steps of the approach. We believe this behaviour (jumping a step) is avoided by the solution tree graphical representation and the forms that the group is supposed to fill out with their answers in every step. However, if the learners still jump a step LeCS intervenes notifying the learners. An example of an intervention about jumping a step is "You have just jumped step x".

Sentence Openers

The chat tool (cf. Figure 4) included in the LeCS user interface includes a list of sentence openers. Hence, each time the learners enter a contribution in the chat, they can initiate a sentence using the sentence openers available (that is, the sentence openers use is not compulsory).

Johnson & Johnson (1994) identified sentence openers and specified the most basic and important interpersonal and small groups skills in discussions. They categorised and associated these skills with sentence openers. These categories are: i) communication skills, ii) trust skills, iii) leadership skills, and iv) creative conflict skills. Based on this work, McManus and Aiken (1995) implemented the sentence openers in an intelligent collaborative learning system with the intention to give an example of a particular skill, according to these categories. In this system, the students sent messages to each other by selecting a sentence opener from a menu, and then elaborated on this sentence opener with additional text. There was a one to one correspondence between the sentence openers implemented by McManus and Aiken and the skills identified by Johnson and Johnson. An ITS offered advice and feedback on the students' skill use during the course of the discussion, and generated feedback at the end of the discussion. The tutoring system's suggestions were based on patterns of interaction denoted by the use of the sentence openers. Likewise, in Robertson *et al*. (1998) the sentence openers are meant to give an example of a particular skill, according to these categories.

In LeCS the sentence openers are just intended to facilitate the discussion. Nineteen sentence openers were selected from the ones used in Robertson *et al*. (1998). This selection was based on the identification of the sentence openers in the dialogues (that is, in the chat log files) of the empirical study.

A Typical Session using LeCS

A typical session using LeCS starts with the learners registering as a member of a group that will work on a particular case study. After the learners join the session, this case study is enabled, whereas the other case studies included in the case library are disabled during this session. Also, the group is supposed to choose one of its members to be the group coordinator. Next, the learners start working by browsing the web pages that include: the instructions of how to proceed to use the system, the case study contents, and the steps of the approach. Afterwards, in all the steps described in Table 1 the following procedure takes place:

- First, the learners individually answer the question posed in the step, using the text editor tool. This activity might be carried out off-line, asynchronously, specially in a distance learning context. In this phase of the case solution development process, *individual learning* takes place.
- Then, the learners work collaboratively, that is, they discuss to reach a consensus about the group's joint answer to the step question. The starting point for the discussion is the differences or similarities between the group members' individual answers. These answers can be copied from the text editor and pasted into the chat in order to be made available to the group. This *collaborative learning* phase is carried out on-line, in a synchronous mode and all the group members should be logged on.

- Once the group reaches an agreement about what to respond to the present question, the group coordinator fills out the form with the *step answer* agreed on by all the group members. This answer represents the accomplishment of the activity demanded in that particular step. The system reasoning is based on this group answer rather than on the individual ones.
- After that, the group moves on to the next step, in which the previous procedure (individual learning, collaborative learning, and group agreement on a joint answer) takes place again. By proceeding sequentially through the steps, the learners will be guided in the development of the case study solution.

DISCUSSION

In this session we discuss some issues, related either to the LeCS implementation, or to the distance learning from case studies model that we have proposed, or both. These issues concern the case solution graphical representation, the synchronicity and non-synchronicity in distance learning, the case solution evaluation in LeCS, and finally case-based instruction.

Case Solution Graphical Representation

In LeCS we represent the case study solution development according to the Seven Steps approach (Easton, 1982) as a tree, and display it graphically to the learners on the user interface. In the empirical study, the subject pairs often forgot about a branch of the tree or even missed a step during the solution development, thereby, in our opinion, decreasing the quality of this process. Hence, the aim of this representational tool is that the learners could make more informed and perhaps better decisions along the process of development of the case solution.

However, there are some issues concerning this representation that should be made clear. The effectiveness and efficiency of this representational tool was not tested. The mapping across representations (that is, the group answers as the nodes of the tree and the arcs as the relations between those answers) from the learners' point of view is only potential so far. Although we had significant experimental evidence for proposing this kind of representation it is still not clear that the students will manage to relate their dialogue contributions to what is represented on the user interface.

Synchronicity and Non-synchronicity in Distance Learning

The web, when used as a medium for distance learning, usually allows the non-synchronicity of the learning processes when a group of learners is attending a distance course. In our proposal of Distance Learning from Case Studies (Rosatelli & Self, 1999), a typical session to solve a case study at a distance is envisaged as having both individual learning and collaborative learning. As a result, the learners are constrained in time in the sense that at some points they will have to get together to have an on-line discussion.

The interleaving of individual learning and collaborative learning aims to tackle this problem. Although collaborative learning is the system's main focus, individual learning plays an important and essential role in this learning process allowing, to a certain degree, asynchronous distance learning.

Case Solution Evaluation

One issue that is not directly addressed by LeCS is the evaluation of the case study solution. The evaluation remains as a case instructor task and, as such, most times it depends heavily both on his or her teaching style and on the learning objectives of a particular case study.

In order to support the case solution evaluation, LeCS provides the case instructor with information to make a judgement, maybe even better than if he or she were teaching with case studies in the traditional classroom. As we stated above, LeCS is able to handle a group of distant learners, taking into account the learning processes going on in each individual learner. It keeps records of the individual contributions both quantitatively and qualitatively (for instance, participation regarding the number of contributions and the contributions themselves), of the time the group took to develop the case study solution, of the misunderstandings concerning the case study, of the difficulties (or the absence of them) about the underlying knowledge that the learners should have acquired previously and that the solution of the case requires, and so forth. Additionally, all the learners and group interactions with LeCS are logged. Thus, the case instructor can make use of these records to analyse the processes of reaching an agreement and the group decisions at different points of the solution as well as the final case study solution.

Case-based Instruction

LeCS makes use of cases, namely case studies, but it does not constitute an application of case-based reasoning (Kolodner, 1993) to computer-based learning. Instead, it applies the Learning from Case Studies educational method, as it is most typically used, to a computer-based system that allows distant learners to work collaboratively on the solution of a case study. The main focus of LeCS is collaborative (distance) learning and all the issues that arise from the application of the case method in this learning scenario.

CONCLUSIONS AND FUTURE WORK

The first stage of the LeCS implementation is concluded. Based on the recommendations of the empirical study that we carried out we implemented a first prototype of the system. The next stage is to test LeCS with pairs of subjects in an experiment along the same lines as the empirical study (Rosatelli & Self, 1999). The intention is to evaluate the system's performance and the effectiveness of its tools and functions, particularly the case solution graphical representation, having this previous empirical study as a parameter of comparison. At this stage we also intend to carry out a study on the LeCS user interface design. This study goal is to test the appropriateness of the user interface, which is related to how intuitive it is, and how easy it is for the user to learn how to use it.

In the next version of LeCS we plan to make the adjustments indicated by these experimental results. In addition, we intend to tackle issues such as the recognition of many solution paths by the learner, the student modelling, and the case study modelling.

Concerning the solution paths, despite the fact that the mapping of the solution process development made by the system might be used for recognising possible solution paths, LeCS does not implement this function. The solution path recognition and evaluation is still a task of the case instructor in the system currently implemented. In a future version, as a result of the interaction with many groups of students solving the same case study, the system should be able to store the solution paths proposed by different groups. This information could be used by the system to carry out the solution path recognition.

Also, from the point of view of student modelling, in LeCS current version, the case solution and all the files available for the case instructor to carry out an evaluation concerning the domain refer to the group rather than the individual students. In the next version of LeCS we intend to check the individual contributions and responses. That would be an approach to build individual student models. Thus, the teacher would be able to evaluate the students' performance individually too.

Regarding the case study modelling, in order to model the interventions about case-specific utterance patterns, LeCS provides a constraints editor for the case instructor. The constraints editor allows the case instructor to edit the set of constraints to a case study (cf. Case-Specific Utterance Patterns subsection). However, it allows the modelling of only one kind of sentence or word to each constraint. If the case instructor intends to contemplate different words that the students might use, he or she should edit additional constraints. The justification for using this kind of approach in the current version of LeCS is that we observed in the empirical study that the students' vocabulary used in the case solution is usually restricted to the words used in the case study text. The next version of the system should provide the possibility to model more flexible constraints, through the implementation of logic operators such as *and* and *or*.

Finally, after making the necessary adjustments and changes, the following stage is to test LeCS with more than one group of learners working simultaneously rather than only a pair of subjects at a time. The final evaluation stage is to use LeCS in a real distance learning context.

ACKNOWLEDGEMENTS

We thank the empirical study volunteers.

REFERENCES

Burt, G. (1997). *Face to Face with Distance Education*. Milton Keynes, UK: Open and Distance Education Statistics.

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier & A. Lesgold (Eds.) *Intelligent Tutoring Systems* (pp. 261-269). Lecture Notes in Computer Science Vol. 1086. Berlin: Springer-Verlag.

Christensen, C. R. & Hansen, A. J. (1987). Teaching with cases at the Harvard Business School. In C. R. Christensen & A. J. Hansen (Eds.), *Teaching and the Case Method: Text, Cases, and Readings* (pp. 16-49). Boston, MA: Harvard Business School.

Easton, G. (1982). *Learning from Case Studies*. London: Prentice Hall.

Franklin, S. & Graesser, A. (1997). Is it an agent or just a program? A taxonomy for autonomous agents. In J. Müller, M. J. Wooldridge & N. R. Jennings (Eds.) *Intelligent Agents III* (pp. 21- 35). Lecture Notes in Artificial Intelligence Vol. 1193. Berlin: Springer-Verlag.

Genesereth, M. R. & Ketchpel, S. P. (1994). Software agents. *Communications of the ACM*, 147, 48-53.

González, M. A. C. & Suthers, D. D. (2000). A coached collaborative learning environment for entity-relationship modeling. In C. Frasson, G. Gauthier & K. VanLenh (Eds.) *Intelligent Tutoring Systems* (pp. 324-333). Lecture Notes in Computer Science Vol. 1839. Berlin: Springer-Verlag.

Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A., & Vassileva, J. (1998). The intelligent helpdesk: Supporting peer-help in a university course. In B. P Goettl, H. M Halff, C. L. Redfield & V. J. Shute (Eds.) *Intelligent Tutoring Systems* (pp. 494-503). Lecture Notes in Computer Science Vol. 1452. Berlin: Springer-Verlag.

Hansen, A. J. (1987). Suggestions for seminar participants. In C. R. Christensen & A. J. Hansen (Eds.), *Teaching and the Case Method: Text, Cases, and Readings* (pp. 54-59). Boston, MA: Harvard Business School.

Herreid, C. F. (1996). Case study teaching in science: A dilemma case on "animal rights". *Journal of College Science Teaching*, 25, 413-418.

Johnson, D. W. & Johnson, R. T. (1994). *Learning Together and Alone*. Englewood Cliffs, NJ: Prentice Hall.

Kolodner, J. L. (1993). *Case-based Reasoning*. San Mateo, CA: Morgan Kaufmann.

Labrou, Y. & Finin, T. (1997). A proposal for a new KQML specification. *Technical Report CS-97-03*, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, USA.

Lester, J. C., Converse, S. A., Stone, B. A., Kahler, S. E., & Barlow, S. T. (1997). Animated pedagogical agents and problem solving effectiveness: A large scale empirical evaluation. In B. du Boulay & R. Mizoguchi (Eds.) *Artificial Intelligence in Education* (pp. 23-30). Amsterdam: IOS Press.

McManus, M. M. & Aiken, R. M. (1995). Monitoring computer-based collaborative problem solving. *Journal of Artificial Intelligence in Education*, 6(4), 307-336.

Meyers, C. & Jones, T. B. (1993). *Promoting Active Learning: Strategies for the College Classroom*. San Francisco, CA: Jossey-Bass Publishers.

Mitrovic, A. & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education*, 10, 238-256.

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

NCSA (1998). The NCSA Habanero Users Guide. Available at http://havefun.ncsa.uiuc.edu/habanero/Docs/index.html

Nwana, H. S. (1996). Software agents: An overview. *The Knowledge Engineering Review*, 11(3), 205-244.

Oram, I. (1996). Computer support of learning from cases in management education. *Innovations in Education and Training International*, 33(1), 70-73.

Parkes, A. P. & Self, J. A. (1990). Towards "interactive video": A video-based intelligent tutoring environment. In C. Frasson & G. Gauthier (Eds.), *Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education* (pp. 56-82). Norwood, NJ: Ablex Publishing Corporation.

Preston, J. A. & Shackelford, R. (1998). A system for improving distance and large-scale classes. In *Proceedings of Third Annual Conference on Integrating Technology into Computer Science Education* (pp. 193-198). New York: ACM Press.

Rich, M. (1995). Supporting a case study exercise on the World Wide Web. In D. Jonassen & G. McCalla (Eds.) *Proceedings of the International Conference of Computers in Education* (pp. 222-228). Charlottesville, VA: AACE.

Robertson, J., Good, J., & Pain, H. (1998). BetterBlether: The design and evaluation of a discussion tool for education. *International Journal of Artificial Intelligence in Education*, 9, 219-236.

Rosatelli, M. C. & Self, J. A. (1998). An empirical qualitative study on collaborating at a distance to solve a case study. *Technical Report 98/27*, Computer Based Learning Unit, University of Leeds, UK.

Rosatelli, M. C. & Self, J. A. (1999). Supporting distance learning from case studies. In S. P. Lajoie & M. Vivet (Eds.) *Artificial Intelligence in Education* (pp. 457-564). Amsterdam: IOS Press.

Rosatelli, M. C., Self, J. A. & Thiry, M. (2000). LeCS: A collaborative case study system. In C. Frasson, G. Gauthier & K. VanLenh (Eds.), *Intelligent Tutoring Systems* (pp. 242-251). Lecture Notes in Computer Science Vol. 1839. Berlin: Springer-Verlag.

Roschelle, J. & Teasley, S. (1995). The construction of shared knowledge in collaborative problem solving. In C. E. O'Malley (Ed.), *Computer Supported Collaborative Learning* (pp. 69-100). Berlin: Springer-Verlag.

Russell, S. J. & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall.

Schank, R. C. & Abelson, R. P. (1977). *Scripts, Plans, Goals and Understanding.* Hillsdale, NJ: Lawrence Erlbaum.

Shaw, E., Ganeshan, R., Johnson, W. L., & Millar, D. (1999). Building a case for agent-assisted learning as a catalyst for curriculum reform in medical education. In S. P. Lajoie & M. Vivet (Eds.) *Artificial Intelligence in Education* (pp. 509-516). Amsterdam: IOS Press

Shulman, L. S. (1992). Toward a pedagogy of cases. In J. H. Shulman (Ed.), *Case Methods in Teacher Education* (pp. 1-30). New York, NY: Teachers College Press, Columbia University.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley.

Tersine, R. J. (1988). *Principles of Inventory and Materials Management*. 3rd Ed. New York, NY: North-Holland.