A COLLABORATIVE FRAMEWORK TO SUPPORT DISTANCE LEARNING IN PRODUCTION ENGINEERING

Marcello Thiry[1]
Marta Costa Rosatelli[2]
Ricardo Miranda Barcia[2]

[1]Universidade do Vale do Itajaí - UNIVALI/Campus São José
Grupo de Pesquisa em Tecnologia e Sistemas- UNITEC
Rodovia SC 407, km 4, São José-SC, 88122-000 Brasil
thiry@sj.univali.rct-sc.br

[2]Universidade Federal de Santa Catarina
Programa de Pós Graduação em Engenharia de Produção e Sistemas
Caixa Postal 476, Campus Universitário, Trindade, Florianópolis-SC, 88040-900 Brasil
marta@eps.ufsc.br, rbarcia@eps.ufsc.br

ABSTRACT: Supporting Distance Learning in Production Engineering involves enabling collaboration between two or more learners in a distributed environment that runs over a computer network. This implies in both providing a computational framework for collaborative learning and supporting the learning process in a different kind of media. The computational framework for collaborative learning consists of a system that includes the necessary tools for the learners to carry out the required learning activities. Supporting the learning process means providing some assistance to the learners, concerning the functions which are performed by the teacher in the traditional classroom according to the educational method used. In this paper we present our computational, Artificial Intelligence-based approach to the design of a collaborative framework to support distance learning from case studies in production engineering. The context for this application 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.

KEYWORDS: Engineering education, distance learning, artificial intelligence in education.

RESUMO: O suporte ao ensino de Engenharia de Produção a distância envolve possibilitar a colaboração entre dois ou mais estudantes num ambiente distribuído que roda sobre uma redes de computadores. Isso implica tanto em prover uma estrutura computacional para o aprendizado colaborativo assim como dar suporte ao processo de aprendizado num tipo diferente de mídia. A estrutura computacional para o aprendizado colaborativo consiste de um sistema que inclui as ferramentas necessárias para os estudantes desenvolverem as atividades de aprendizado requeridas. O suporte ao processo de aprendizado significa dar alguma assistência aos estudantes, no que se refere as funções desempenhadas pelo professor na sala de aula tradicional de acordo com o método educacional escolhido. No presente artigo apresentamos nossa abordagem computacional, baseada em Inteligência Artificial para o *design* de uma estrutura colaborativa de suporte ao ensino a distância de Engenharia de Produção usando o método de estudos de casos. O contexto para esta aplicação são cursos a distância de Engenharia de Produção que incluem disciplinas que apresentam problemas sem solução ótima e com mais de uma solução satisfatória sobre design, análise, seleção, planejamento e/ou situações de decisões da administração. Geralmente tais problemas são derivados da experiência real, refletindo situações reais pelas quais passam engenheiros e gerentes, e são usados para treinar os estudantes para a prática profissional.

1 INTRODUCTION

The development of new technologies for training and education and the learning requirements demanded by the dynamics of modern society make of distance learning (Moore & Kearsley, 1996) a timely issue, which major point is to enable the expansion of the classroom into a broader universe, allowing people to educate themselves in any place at any time. In this sense, the World Wide Web, together with the Internet, constitutes an unrivalled environment for distance education. The development of uncountable Web-based educational applications and Web-based courses for distance learning over the last few years is a demonstration of this reality. However, there are issues that should be addressed in the design of distance education. Porter (1997) points out characteristics that must be taken into account when planning a virtual classroom, regardless of the technologies involved. Among those characteristics are availability, and ease of use and location of the appropriate tools, permitting the learners to be free to experiment, test their knowledge, accomplish tasks, and apply what was discussed or read. We believe these features evoke, in a computer-based environment, a collaborative system where learners can interact in the discussion of ideas and learn by experience.

Supporting Group Activity in Distance Learning from Case Studies (Rosatelli, 1998) involves enabling collaboration between two or more learners in a distributed environment that runs over a computer network. *Learning from Case Studies* (Christensen & Hansen, 1987) is a well established educational method that has been widely used for many years, in diverse fields and particularly in engineering education. However it has been overlooked in what concerns its application to distance education and to systems using the new communications technologies. The present work consists of an application that aims to “bridge this gap” as it allows case-based instruction to take place in a Web-based system.

In order to accomplish this, it is necessary to both provide a computational framework for collaborative learning and assist the learning process in this kind of media. The former implies in allowing collaboration between the learners, what is needed in Learning from Case Studies as the method usually is applied to groups. That is, collaborative learning, defined as “a coordinated synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem” (Roschelle & Teasley, 1995), fits perfectly well to the kind of interactions demanded by the method. Therefore, the computational framework for collaborative learning consists of a system that includes the necessary tools for the group to carry out the required learning activities. The latter, i.e., assisting the learning process, means providing some kind of support to the learners, concerning the functions which are performed by the teacher in the traditional classroom according to the case method .

In this paper we present our computational, Artificial Intelligence-based approach to the design of a collaborative framework to support distance learning from case studies in production engineering. This collaborative framework is agent-oriented (Thiry *et al.*, 1998a; Thiry *et al.*, 1998b) and allows the system to perform functions concerning the assistance to the learners during the collaborative process of developing a solution to case study (Rosatelli & Self, 1998; Rosatelli & Self, 1999).

The context for this research is a distance course on production engineering that includes disciplines containing open-ended engineering 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.

Section 2 presents an overview of case-based instruction, focussing on its important features and their significance to an intelligent system to support collaboration. Section 3 describes our agent-based collaborative framework. Section 4 discusses an application in Production Engineering. Section 5 presents the conclusions and recommendations for further research.

2 CASE-BASED INSTRUCTION

A case can take diverse forms such as a story, an event or a text and is defined by Shulman (1992) as an "instance of a larger theoretical class", "an example of a broader category". It is a member of a class of events and as such it represents certain features of that class. Cases can be used in a range of subjects as varied as law, psychology, psychiatry, architecture, education, engineering, business, management, etc. The common characteristic among those disciplines is that case studies are used as a teaching method where the skills of solving complex and unstructured problems are required. There is no analytical technique or approach to solve this kind of problems what means cases are suitable to open-ended problems that have no "correct" or clear-cut solution. An engineering case is defined as a written account of an engineering activity as it was actually carried out (Kardos, 1979).

The case study will basically furnish raw material for the case discussion, which is a central issue in Learning from Case Studies. 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 - that we adopt in this work - 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 the system so that it can interact with the learners, providing support and feedback during the case solution process. Each step has its own goal and suggests a range of activities to be carried out by the learners in order to achieve such goals (see Table 1). The case discussion in its turn is intrinsically related to the instructor's role in the case method. The leadership of the case discussion process is a critical responsibility of the instructor who, rather than having a substantive knowledge of the field or case problem, must lead the process by which individuals and the group explore the complexity of a case (Christensen & Hansen, 1987).

THE SEVEN STEPS	ACTIVITIES
Step1. Understanding the situation	Relate, summarise
Step2. Diagnosing problem areas	List problems
Step3. Generate 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

Table 1- Adapted from the Seven Steps approach (Easton, 1982)

3 THE COLLABORATIVE FRAMEWORK

The collaborative framework that supports the intelligent learning environment proposed is described in this section. Firstly we present the multi-agent approach used: the kind of agents, how they operate, and its communications architecture. Then, we detail the model of case discussion management - the collaborative process of discussing and solving the case study in a learning scenario of a group of distant learners - within this multi-agent system approach.

3.1 THE MULTI-AGENT APPROACH

The generic model of the collaborative framework includes three kinds of agents defined as follows: interface agent, responsible for the interaction with the participant; information agent, which browsers databases and knowledge bases; and advising agent, which offers personal assistance to participants. Such agents are able to co-operate over a network, whether the Internet, an Intranet, or even isolated local networks (Figure 1).

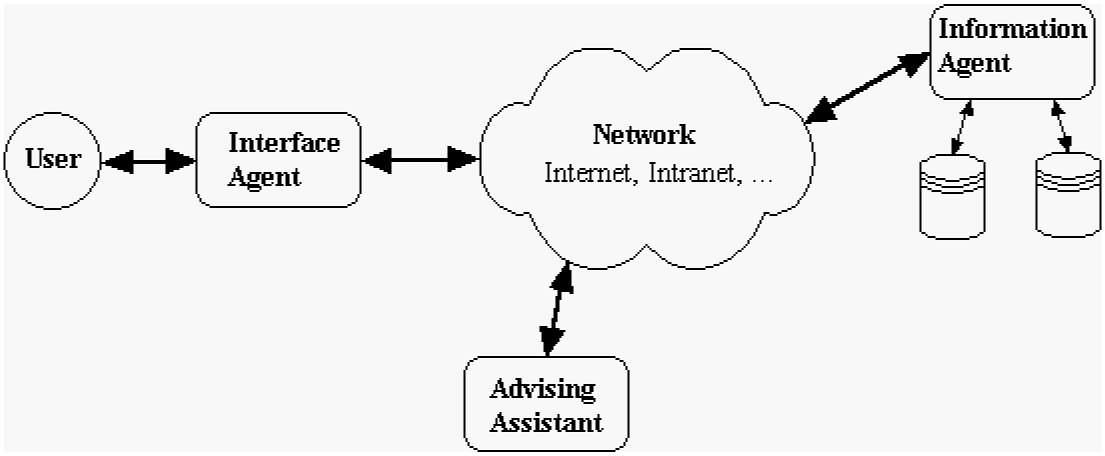

Figure 1. Generic architecture for the collaborative learning environment.

3.1.1 INTERFACE AGENT

The interface agent interacts with the participant and co-operates with the other agents exchanging information on capacities, compromises and learning goals of the participants. In addition, the interface agent must feature the capability to represent its participant, being able to act for the participant in his/her absence.

3.1.2 INFORMATION AGENT

This agent stores the representation of learners' domain interpretations, reasoning with the information passed on by the interface agents. The agents carry out all communication process. Both interface and advising agents can access the information agent.

The information is divided into two different categories: didactic material and knowledge base. Didactic material comprehends HTML pages, images, texts, and multimedia. This category of information is being stored currently in a relational database on an Internet site. The knowledge base is organised by a set of cases and rules. Actually, the current version uses rules only to represent the learners misunderstandings. Cases represent the situations to be explored by the learner.

3.1.3 ADVISING AGENT

The advising agent adopts the paradigm of an intelligent personal assistant. It is, therefore, able to assist the group of learners by constantly monitoring their actions and intervening to indicate a proper application of knowledge. This is achieved through the verification of the group of learners' situation at the moment.

3.2 COMMUNICATION ARCHITECTURE

The framework is based on the structure proposed by Genesereth & Ketchpel (1994), where and architecture for interoperability is defined. Agent communication takes place by a default message exchange using ACL (Agent Communication Language) standard. This standard establishes that communication does not happen directly between agents, but rather through a special program called *Facilitator*. Figure 2 shows the communication structure used. Two local data bases are implemented: in the first, the Facilitator stores all necessary information in order to route messages; in the second data base it logs all the exchanged messages. The format of these data bases is quite simple and it is not relevant to describe them here.

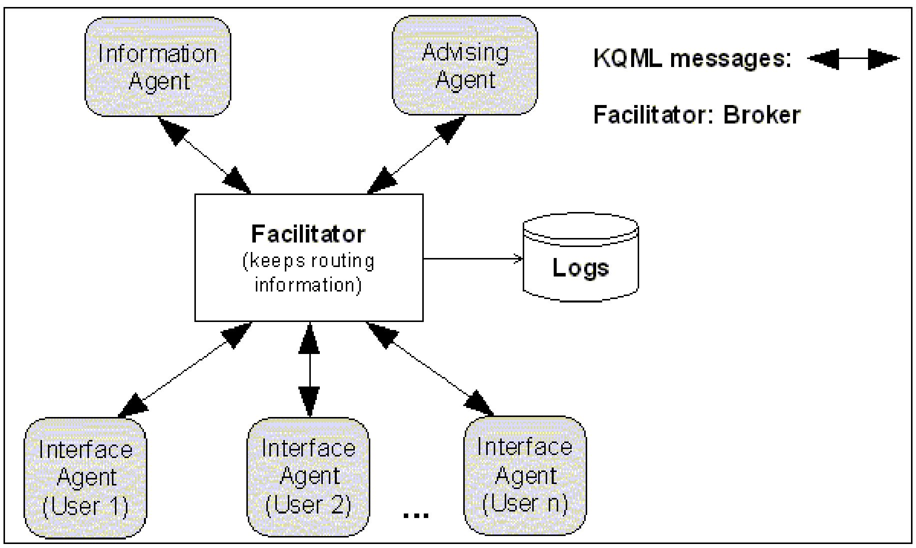

Figure 2. Communication Architecture adopted.

3.3 THE PROPOSED MODEL

In this sub-section we describe the system functions highlighting the AI techniques used within the agent-based approach described above. Firstly we present the domain knowledge base; secondly the dialogue management and thirdly the Learning from Case Studies method knowledge base.

3.3.1 THE DOMAIN KNOWLEDGE BASE

The domain knowledge base concerns the system knowledge about a case study, which is suitable to be represented with scripts (Schank & Abelson, 1977). The scripts are defined as a "standardised general episode" and its knowledge structure abstracts the sequence of events present in the case. The notation used in the representation of the case study as a script are conceptual graphs (Sowa, 1992). In this version, however, only the instructional relevant aspects, i.e. the possible learners misunderstandings about the case (which are judge as so by the case instructor), are represented. In order to do this in addition to the conceptual graphs, constraint-based modelling (CBM) is also used so that the system is able to identify those misunderstandings (Mitrovic & Ohlsson, 1999). A state constraint can be represented as a pair of patterns, which is a list of elementary propositions. Each state constraint is a pair of tests on problem states. The semantics of a constraint is: *if the properties Cr hold, then the properties Cs have to hold also* (or else something is wrong).

3.3.2 THE DISCUSSION MANAGEMENT

The discussion management comprises four main functions to be accomplished: *analysis* (*parsing and semantic interpretation)*, *controlling the violation of constraints*, *controlling the solution path*, and *generating interventions*.

The input for the discussion management are the outcomes of each of the Seven Steps, i.e. the sentences that compose the learner's answers to each step question. This is equivalent to say that the input for the system reasoning is what is going on in the dialogues. Thus, the monitoring of the case discussion demands *parsing and semantic interpretation* (Russel & Norvig, 1995) of the learners utterances contained in the dialogue. The aim is to identify a pattern that might lead to an intervention by the system.

Case-specific learners' utterances may represent points at which the system should provide support, intervening in order to draw the learners' attention to a misunderstanding. This means that for the system to generate an intervention it will be *controlling the violation of the constraints* represented in the domain knowledge base against the learners' case specific utterances. To test weather a given problem state is consistent with a set of constraints, the procedure is to compare the state against all constraints and noticing any constraints violations. This is done in two steps. Firstly all the relevance patterns are tested against the problem state to identify those constraints that are relevant in that state. Secondly, the satisfaction pattern of only the relevant constraints are tested against the problem state. If the satisfaction pattern of a relevant constraint matches the current state, then the constraint is satisfied. If the satisfaction pattern of a relevant constraint is not satisfied, then that state violates the constraint. This is the situation that activates the interventions generation module.

In addition, during the case solution development the learners might also miss the interconnections between the answer components through the Seven Steps, therefore losing track of their solution paths. If the Seven Steps are correctly followed, there will be a list of alternative solutions, an outcomes prediction of each alternative solution, and a pros and cons list to each outcome. By *controlling the solution path* the system should be able point out any incoherence concerning the steps' expected outcomes. In order to accomplish this function the system dynamically generates a tree data structure (Russel & Norvig, 1995), a knowledge representation of the case solution development according to the Seven Steps Approach. The inputs are the outcomes of each step, that is, the solution tree is generated from the sentences that compose the learner's to each step question. The resulting tree represents the solution path taken to solve the case study and is referred to as the solution tree. The information about what is required from each step comes from the *Seven Steps Knowledge Base* and corresponds to the learners' answer to a certain step, so that the system knows what step question the learners are answering. The interface agent also provide information about which step the learners are currently working on (which Web page they are accessing). The tree final state is represented then for all (ideally) alternative solutions that can be generated through the the Seven Steps Approach.

Finally, *generating an intervention* concerns promoting and monitoring the case discussion, similarly to the case instructor in the traditional classroom, who intervenes during the case discussion. When making an intervention the system provides some feedback to the learners concerning the case solution. The knowledge in this module is represented by a set of production rules (Russel & Norvig, 1995) that are applied according to the input coming both from the constraints violations and from the interface agents concerning timing , participation or the tools co-ordination.

3.3.2 THE SEVEN STEPS APPROACH KNOWLEDGE BASE

This knowledge base concerns the approach used to guide the case study solution process, the Seven Steps (Easton, 1982). The Seven Steps are represented in this module also with scripts as they are a sequence of activities to be developed. The scripts contain the information about the sequence of steps, including their objectives and correspondent questions. Besides, it also includes information about what is required from the learners in that step in terms of the case solution, i.e., the steps expected outcomes.

4 APPLICATION IN PRODUCTION ENGINEERING

4.1 PRELIMINARY EXPERIMENT

The main objective of the preliminary experiment (Rosatelli & Self, 1998) was to observe how the process of discussion between distant learners collaborating on the solution of a case study occurs. As mentioned above, in Learning from Case Studies as applied in the traditional classroom, the discussion is a main topic. However, little or no information is available concerning the discussion in the application of the case method to distance learning, using networked computers and a collaborative learning environment. Therefore, the study aimed to further investigate the issues that arise from the case discussion, in this kind of medium and environment.

The collaborative learning environment for the experiment was based on NCSA Habanero v2.0 Beta 2 (NCSA, 1998), which is a collaborative framework written in Java. It includes a Client, a Server and a set of applications. Among the available applications set in the current version, the 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. The collaborative aspect in the Web browser means that if one of the participants in a session accesses a certain Web page, the others participating in the same session would view the same page on their screen. Concerning the text editor, the participants can edit a text together, typing one at a time in the same text area.

A typical session to solve the case study included:

- *The browsing of the Web pages*, that displayed all the relevant information to the learners on how to proceed to solve the case according to the Seven Steps approach in this environment (i.e., which tool to use, how, and when). The Web pages design is based on the Seven Steps approach. The first set of pages presents, 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. The second set of pages refer to the Seven Steps. Each page includes a description, the goal, the question and the demanded activity concerning each step. The hyperlinks are supposed to be followed sequentially, according to the nature of the approach. After reading the case, the learners initiate the process of solution, accessing the Web page referring to the first step of the methodology.

- *Individual Learning*, when the learners are supposed to answer the question posed in the step, working individually. They type in their individual answers using the text editor facility.
- *Collaborative learning,* when the learners were supposed to work collaboratively. They are supposed to reach a consensus in order to have a joint answer, having as a starting point the differences or similarities between their individual answers.
- *Case discussion,* when the discussion to reach an agreement about a joint answer takes place, using the chat tool.
- *Case solution, w*hen the learners after reaching an agreement, present their joint answer. After finishing with the current step answer they move on to the next step, where the previous procedure takes place again. By proceeding sequentially through the steps, the learners will be guided to the case study solution.

The case study used in the experiment (Herreid, 1996) didn't require any specialised knowledge in order to be solved and is believed to motivate and encourage discussion, as the subjects wouldn't be attending a course or learning a particular domain, which is the standard situation in the application of the case method. The subjects for this experiment were 5 pairs of postgraduate students, who volunteered to take part in the experiment. After being randomly paired, the subjects were located in separate rooms, using network connected computers (PCs under Windows NT) to collaborate at a distance in the solution of the case. The experiment emphasised the collaborative work in the solution of the case study, as well as the reaching of an agreement demanded in every intermediate decision point (i.e., the answer to each step question) and in the final case solution.

4.2 RESULTS AND IDENTIFIED PROBLEMS

The preliminary experiment proved the feasibility of carrying out a case discussion in a distance learning scenario. The Habanero framework (NCSA, 1998) provided the necessary tools demanded by the case activities to be developed as proposed, that is, collaboratively. Thus, individually, the needed tools were available in the framework: the Web browser, the collaborative text editor and the chat tool.

On the other hand, the experiment served to point out it is not sufficient just to have a collaborative framework that allows collaboration at a distance. It is necessary to provide an application where both the co-ordination between the tools used and the support to the development of the case solution can be properly accomplished. We believe the agent-based architecture described in Section 3.1 allows both functions to be achieved.

In addition, it is necessary to provide an interface that is designed according to those goals. As a result, the observations and results of the experiment concerning the tools use as well as what is proposed in terms of the system intelligence inform the design of the system interface.

4.3 THE PROPOSED INTERFACE

The interface (see Figure 3) for the collaborative framework proposed displays the system graphical components that are described below:

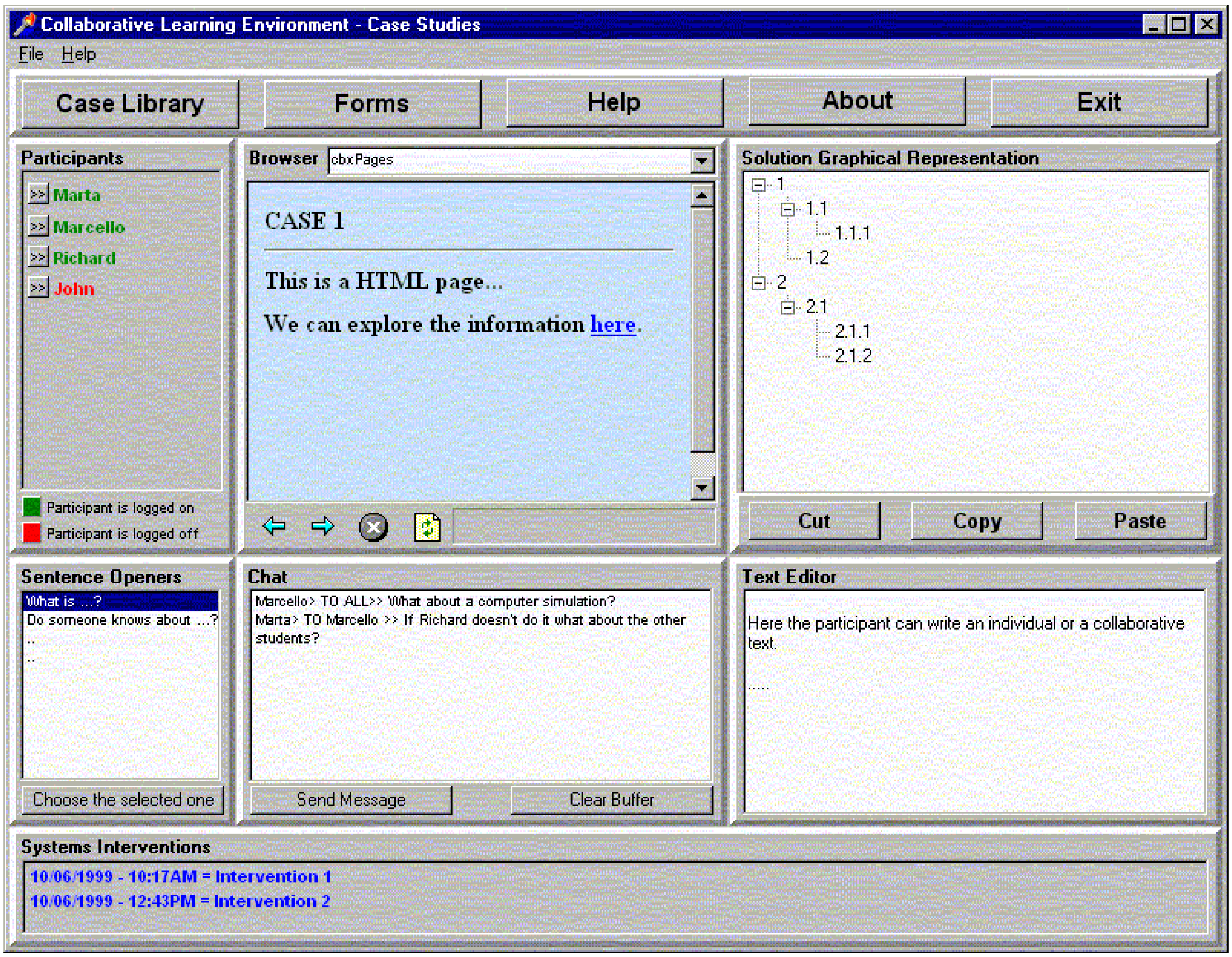

Figure 3. Collaborative Framework Interface

- *Pull-down menu* containing the following elements:

- *case studies library* containing the set of cases available to the learners;
- *forms*, where the agreed joint answers to each step and the solution to the case are typed in. Rather than be parsing all the sentences typed in by the different participants during the discussion, the system only parsers what is presented as the joint agreed answers to each step question and the case solution;
- *on-line help*, containing the relevant disciplines to the solution of that case;
- *about*, and
- *exit*.

- A *participants list,* with the name of the group members both that are logged in and logged of at that moment.
- A *menu of sentence openers*, which objective is to facilitate the reaching of an agreement during the discussion (Robertson et al., 1998). The learners select a sentence opener from a menu and then elaborate on this opener with additional text.
- A *browser*, to access the Web pages containing the learning materials.
- A *chat*, where the discussion takes place
- A *solution graphical representation window*, where the learners can visualise the building up of their solution, which is done step by step, adding in each step to what was done in the previous one. There, the solution tree generated by the system that represents the case solution so far is displayed in order to avoid that the learners miss the interconnections between the answer components and loose track of their solution paths. The graphical elements in this window are editable by the learners when they disagree with the system reasoning (e.g., the link the system makes between a node and its parent node is not what the learners mean in their textual answer).
- A *text editor* (non collaborative), where the learners type in their individual answers. As described in Section 4.1 the learners are supposed to elaborate their individual answers and subsequently, based on the differences/similarities of their answers, discuss their ideas, reach an agreement and provide a joint answer. In order to do this, they had to present each other's their answers. The chat and the collaborative text editor tools are inadequate for this function because as they are collaborative, each individual could interfere with each other answers. For instance, just by clicking on the collaborative text editor window, it is possible to type in this window and

to alter other one's individual answer. The chat tool, on the other hand, does not allow the answer text to be edited. The interface should provide a text editor tool that, despite included in the collaborative framework, would be viewable by all the learners but editable only by "its owner".

- A *system interventions window*, that displays what the system communicates to the learners. The communications are the system's interventions concerning the constraints violations identified in a case-specific utterance that might characterise a misunderstanding about the case solution, denoting difficulties in answering a particular step question. The objective of such intervention is to draw the learners' attention to this potential misunderstanding. Also, the system should intervene (1) controlling the time, when the time spent by the group with on-line collaborative work exceed the time limits; (2) encouraging participation, when there is a higher degree of participation concentrated in one of the peers across so that the other peer(s) offer only a small number of contributions; and (3) co-ordinating the tools use, when the actions initiated by one of the learners in a particular tool window despite being visible to his/her distant peer, is not enough to keep the group co-ordinated concerning switching between tools.

4.4 AN EXAMPLE OF A CASE STUDY APPLICATION IN PRODUCTION ENGINEERING

The context for this application is a distance course on production engineering that includes disciplines containing engineering case studies, i. e. open-ended engineering 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.

An example of the kind of case study that is currently being modelled and can be worked on by a group of learners using the present system is for instance a case study on Just-in-Time (Tersine, 1988, p. 424). A typical session to solve a case study using the proposed system is envisaged as described in Section 4.1, similarly to the preliminary experiment which was carried out, but with the additional functionality and support provided by the system. This functionality and support consists basically of the system feedback concerning the case solution development during a session: the system interventions and the display of the graphical representation of case solution development .

5 CONCLUSIONS AND FUTURE RESEARCH

Bearing in mind what was previously exposed our system tackles basically how to manage the case solution process when a group of learners is geographically dispersed, collaborating to solve a case study. The multi-agent framework supports collaborative learning and is the basis for all the systems functions related to the support of the case discussion.

However, from the perspective of case-based instruction, the issue of retrieving a case, according to the needs of the learner is particularly relevant to this system. From a computational, Artificial Intelligence point of view, case-based learning (Schank, 1991) is defined as the situation when the learners acquire new knowledge from the exploration of situations stored in library of past experiences. In computational systems to support traditional education, the designer generally has the knowledge about the domain and decides how this knowledge should be made available. One of the advantages of applying case-based reasoning is to work with a partially incomplete knowledge of the domain. There is no need for an explicit model of the domain. These characteristics allow that the increase of knowledge be incremental. Furthermore, from the educational standpoint, cases can be explored as situations to be presented to the learners, who will be prompted to find solutions from similar cases, synthesise them, apply them in this new solution and then furnish an explanation for the choice. These aspects make case-based learning an interesting tool for learning/teaching processes.

In conclusion, the present system can be extended to be used as in (Thiry , 1999), adding to the present system all the functionality of case-based learning. In this case, the resulting system in a first level would perform the retrieving of a case according to the teaching/learning objectives and all the interactions learners/system that might take place concerning this process. On a second level, the case discussion management as it is proposed here, i. e. the collaborative process of discussing and solving the case study in a learning scenario of a group of distant learners with the support of the system, takes place.

REFERENCES

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

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

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

Hansen, A. J. (1987). Suggestions for Seminar Participants. In C. R. Christensen with A. J. Hansen, *Teaching and the Case Method: Text, Cases, and Readings*, pp. 54-59. Boston, Massachusetts: Harvard Business School.

Herreid, C. F. (1996). A Dilemma Case on “Animal Rights”. *Journal of College Science Teaching*, 25, 413-418.

Kardos, G. (1979). *Engineering Cases in the Classroom.* Available at http://www.civeng.carleton.ca/ECL/cclass.htm

Mitrovic, A. & Ohlsson, S. (1999). Evaluation of a Constraint-Based Tutor for a Database Language. *International Journal of Artificial Intelligence in Education*, 10, to appear.

Moore, M. G. & Kearsley, G. (1996). *Distance Education: A Systems View*. Belmont, CA: Wadsworth Publishing Company.

NCSA (1998). *NCSA Habanero Environment*. Available at http://havefun.ncsa.uiuc.edu/habanero/.

Porter, L. (1997). *Creating the Virtual Classroom: Distance Learning with the Internet.* Wiley Computer Publishing, NY: Wiley & Sons.

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. (1998). *A Web-based System to Support Group Activity in Distance Learning from Case Studies*, First-Year Report, Computer Based Learning Unit, University of Leeds.

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.

Rosatelli, M. C. & Self, J. A. (1999). Supporting Distance Learning from Case Studies. Paper accepted to the AIED’99, 9th International Conference on Artificial Intelligence in Education to be held on July 19-23, 1999 in Le Mans, France.

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. Heidelberg: Springer-Verlag.

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

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

Schank, R. (1991). *Case-Based Teaching: Four Experiences in Educational Software Design*. Technical Report No. 7, Institute for the Learning Sciences.

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. (1992). Conceptual Graphs Summary. In T. Nagle, G. Nagle, L. Gerholz & P. Eklund (Eds.), *Conceptual Structures: Current Researches and Practice*, pp. 3-52. Ellis Horwood.

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

Thiry, M., Barcia, R. M. & Khator, S. (1998). Personal Assistance: An Intelligent Hybrid System Approach. In *Proceedings of Industrial Engineering Research Conference*. Alberta, Canada.

Thiry, M., Barcia, R. M., Khator, S. & Martins, A. (1998). Intelligent Agent-based Approach for Distance Learning. In *Proceedings of International Conference on Engineering Education*. Rio de Janeiro, Brazil.

Thiry, M. (1999). *Agent-Based Architecture to Support Distance Learning*. Ph.D. Thesis, Graduate Program in Production Engineering, Federal University of Santa Catarina, Brazil.