DOCUMENT RESUME

ED 428 732 IR 019 393

AUTHOR	Verdejo, M. F.; Barros, B.; Abad, M. T.
TITLE	Supporting Distance Learners for Collaborative Problem Solving.
PUB DATE	1998-06-00
NOTE	7p.; In: ED-MEDIA/ED-TELECOM 98 World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications. Proceedings (10th, Freiburg, Germany, June 20-25, 1998); see IR 019 307. Figure may not reproduce clearly.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Case Method (Teaching Technique); *Computer Assisted Instruction; *Computer Interfaces; Computer Managed Instruction; Computer Mediated Communication; *Cooperative Learning; Courseware; *Distance Education; Educational Technology; Foreign Countries; Higher Education; *Instructional Design; Learning Processes; Models; *Problem Solving
IDENTIFIERS	Collaborative Learning; *Learning Environments

ABSTRACT

This paper describes a computer-supported environment designed to facilitate distance learning through collaborative problem-solving. The goal is to encourage distance learning students to work together, in order to promote both learning of collaboration and learning through collaboration. Collaboration is defined as working together on a common problem, communicating, and coordinating activities towards a shared goal. A system is proposed embedding three models: an activity model, a domain model, and a conversational model. The system can support asynchronous collaboration in a number of dimensions: giving structure to activities and communication to articulate the problem-solving task; supplying a share space to build jointly common knowledge; providing relevant know-how by case and techniques libraries; and facilitating further reflection on the whole learning process. Preliminary results indicate that this approach is promising regarding establishing collaborative distance frameworks to enhance students' learning processes. Topics discussed include: features of the distance learning scenario, including institutional framework, kind of scenario, learners, settings, and facilities; the modeling approach used; and developing an application, including the activity model, dialogue model, domain model, and learner interface. A table presents the definition of specification as a complex activity; and a figure illustrates the learner interface. (Author/DLS)

Supporting Distance Learners For Collaborative Problem Solving

M.F.Verdejo, B.Barros
Dep. de Ingenieria Electrica, Electronica y Control, U.N.E.D. Aptdo 60149, Madrid, 28080 Spain
M.T.Abad
Dep. de Lenguajes y Sistemas Informaticos, U.P.C. Pau Gargallo 5, Barcelona, 08028 Spain

Abstract. This paper describes a computer-supported environment to facilitate distance learning through collaborative problem-solving. Our goal is to encourage distance learning students to work together, in order to promote both learning of collaboration and learning through collaboration. Collaboration means working together on a common problem, communicating and coordinating activities towards a shared goal. We propose a system embedding three models: an activity model, a domain model and a conversational model. The system can support asynchronous collaboration in a number of dimensions: giving structure to activities and communication to articulate the problem solving-task, supplying a share space to build jointly common knowledge, providing relevant know-how by case and techniques libraries and facilitating further reflection on the whole learning process. Preliminary results have convinced us that this approach offers promises to establish collaborative distance frameworks to enhance students learning processes.

1. Introduction

Teaching and learning require a range of approaches to suit diverse domains and students. At the root of science courses, including computer science and engineering, is the acquisition of problem-solving skills, usually through practical activities performed individually or in group work. In the case of distance learning situations, communication technology opens for the first time, the possibility of deploying problem-driven and collaboration learning strategies, so that students could acquire and practice a variety of interpersonal and reasoning skills. Recently, several research teams work on developing tools to support the performance of (learning) tasks to be carried out together by students located at different places. As many authors point out, two key ideas are (1) learning activities should be structured to focus students on the issues to learn, and (2) educational software systems should exploit their interactive potential and the capacity to include and manipulate explicit models, in order to stimulate students' knowledge construction processes. Distance collaborative learning can be shaped in many ways [Collis 94], [CACM 96], [Verdejo 96], joint projects is appealing for a range of scientific and engineering subjects. To mention just a few of the current educational projects exploring this area, [Edelson et al. 94] offer a shared Notebook with a set of page types corresponding to a task model of scientific inquiry to be used for collaborative science learning. The page types include questions, conjectures, evidence for and against, information, commentary as wells as plans and their steps. Students can create instances of these pages and link them as their work progresses. A learner interface built upon their scientific inquiry model, suggest students next steps to pursue. [Wan & Johnson 94] report on CLARE, an environment supporting the task of learning to understand scientific literature. They define an explicit process model: summarization, evaluation, comparison, argumentation and integration and provide representations and structure for each of these individual and group activities. Learning as collaborative knowledge construction can be difficult to achieve, among other reasons because often students have to learn at the same time the abilities to collaborate. However, group competencies are basic skills to be acquired not only for the purpose of the learning process at hand, but for future professional proficiency. This paper describes our approach to build a computer-supported environment to facilitate distance learning through collaborative problem-solving. The paper is organized as follows: Section 2 sketches the distance learning scenario. Section 3 gives a description of our modeling approach. An application is presented in section 4, and finally some conclusions are given in section 5.

2. Scenario

A scenario, described as a summary of situations, including information about its setting, participants, their tasks, roles and interactions, is often used to develop a practical understanding of how an end-user will perceive and use a technology-supported environment. For designers, it is a means to perform a joint visualization of candidates ideas in order to explore, refine, and test them for further implementation. In our case, the scenes concern learners-related actions and interactions, embedded however in a broader context. Main features are the following:

• Institutional framework: The institution is a distance learning university, offering diplomas, in a regulated course time frame, with formal examinations. The prevalent model is independent self-study with appropriate learning materials. The use of a collaborative environment should be on a volunteer base (both from academic and student side). Groups using the environment should follow the same calendar (for deadlines, assignments,..) and academic requirements as the rest of students

• Kind of scenario: Collaborative learning, focusing on group project development. Groups of distance learning students, each group has to carry out a project using a system to support their personal and group work. This support should include, in an integrated way: (1) private and shared workspaces with notations for expressing alternative partial solutions at an adequate level of abstraction, (2) facilities to use domain-specific tools, (3) facilities to access information sources in a way relevant to the task at hand, (4) group communication facilities helping to manage the complexity of the interaction.

• Learners, organized in small groups, in each group there are a number of roles: moderator, contributor, and secretary. For the moderator role, two kind of functions are expected: organizational facilitation (such as conduct and end discussions), and social facilitation (motivate and promote cooperation). The role of secretary is to focus on intellectual facilitation, such as summarize contributions or highlight conflicting opinions.

• Settings, described by spatial and time dimensions: Students, Tutors, and Professors are in separate, distant and disperse locations, usually with different schedules, so that interaction and collaboration should be in deferred (no real) time.

• Facilities, comprising conventional and technological material and devices. We assume each participant has access to a standard multimedia computer connected to a central server, using low-cost communication technology (mainly asynchronous electronic mail, computer conferencing, or asynchronous access to file systems)

3. Modeling the system

Our system architecture to support learners for collaborative project development consists of three independent but interrelated components, which are based on the following models: (1) an explicit model of the problem-solving activities or activity model. This includes a description of each activity, to specify whether is a personal one or in group, main goal to attain, kind of information available and possible restrictions. Relationships between activities are also expressed. The group organization (participants and roles) is also provided; (2) a model of the participants interaction, both human-machine and human-human communication. Communication is performed through semi-structured messages using a conferencing system, each message belongs to a larger discourse unit, and depending on its type and purpose is related either to events of the activity model or to objects of the domain model; These two models are inspired on proposals from the CSCW stream, [De Cindio et. al. 88], [Malone & Fry 92] implemented for teamwork in professional environments; (3) a model of the domain comprising two sources of case-based structured knowledge, and a workspace offering a notation to express students ' constructions for a solution.

In our system these three components are integrated and related one to each other. For instance, the activity to be performed can constrain the type of dialogue that is used; an activity is developed linked to a type of problem-solving task in a particular knowledge domain. In this way actions and interactions in the learning process are properly recorded. Thus, not only the whole process is also an inspectable object to reflect upon, but its explicit representation allows, as well, to include supportive actions from the system. The activity model is generic, that is, its elements are valid for any type of activity not necessary educational. The proposed model of dialogue for a moderated debate (structured in terms of sessions, rounds, and messages) is valid for any discussion that has as an objective to build an agreed solution. In the same vein, the notation for expressing domain problem's description, is also generic (objects and relationships to which the user can associate the desired semantic behavior). To combine domain, activity and group modeling in a learning environment aims at least at two purposes, one to facilitate reflection so that students could have the opportunity to consider not only their outcomes but also the process they are following and, the other one, to provide students with support for building and exploiting cumulative understanding, giving them a further motivation: the opportunity to contribute to the extension of the domain memory for the use of future learners. Grounded on these models, different applications can be built following an user-centered design methodology as initially proposed by [Norman & Draper 86] and further developed by [Gould & Lewis 87]. An *iterative*, *formative*, and *situated* design process. Initial analysis leads to a preliminary specification to build a prototype, the evaluation of that release is used as a formative (information gathering) step, providing feed-back for further re-specification and re-implementation. Final testing should be situated with real users on a realistic task.

4. Developing an Application

We started focusing on the learning of a class of problem-solving methods, in particular related to software design. Taking into account our teaching experience on programming projects development for undergraduate students, a first set of scenes where defined, and experiments with small groups of students were conducted. Each scene characterized in terms of participants and task at hand, focused on a main issue to test a range of support tools. Taken all together, had lead to gain insights for an integrated environment. Through observation and discussion of students' performance and results, the notation used to describe domain objects, including the description of problems, requirements for a solution, and the expression of partial designs was revised. Some changes affected the groupwork process model, and so the definition of the private and shared workspaces and their support tools. After this phase, appropriate features for an integrated learner interface have been established, and a prototype built.

4.1. Activity Model

This section sketches the main objects provided by the activity model. An Activity has a name, a description, a mode (either individual or collaborative), participants, a characterization of the information required as data or generated in each activity, as well as a description of the tools needed to carry it out. A part of this information, but properly distinguished, is the result or product. A Complex_Activity is a set of activities with a partial order defined, it can be visualized as a labeled directed graph in which the nodes are activities. Attached to the nodes and/or edges, a control descriptor represents requests that the system will perform automatically during the development of an activity. For example, when the third step would be finished the teacher could establish that he wants to receive a report. Then the system will generate a message to the students requesting the report. A special case of complex_activity is when the set of activities is ordered i.e. a sequence of activities. For each activity there is a descriptor pointing to the set of electronic messages that students have interchanged while carrying on the activity. This descriptor establishes a link between the activity model and the dialogue model. For our scenario, we have written the description of a main activity related to computer programming design. Students have to learn to characterize a problem and then select an approach to solve it using a set of known techniques. As a final result of this activity students have to build a piece of software and a documentation. The design methodology they have to acquire includes the following sub-tasks: Specification, Instantiation, Structured Design, Implementation/ Test, and Generation of the Documentation. Each one is in turn, a complex activity with its own input and output, its time schedule and participants. An outline of the definition for the Specification is showed in table 1.

The individual phase for making a Specification consists on three activities, namely, problem description, selection of an algorithmic technique and making a justification. The description and the selection can be done at the same time; but the justification has to be done once the other two are finished (these restrictions are expressed in a graph, (see figure 1). A tool allows creating and editing an activity model. Along the learning process, the system works with the model, linking dynamically references to the event instances been created. A visualization of the current state of the activity model is available from the learner interface, so that always students can inspect the structure of the whole process, accessing to the information in a structured way, and being aware of the active sub-task in which they are involved.

Specification is a complex-activity consists of (CA1, CA2)			
ACTIVITY	TYPE	P_ORDER	SUBACTIVIES

CA1 is a complex_activity	Individual	graph1	Build Problem description, Decide which algorithm techniques are candidates to solve the problem, Justification of the proposed techniques depending on the chosen description
CA2 is a complex_activity	Collaborative	moderated debate	Mutual understanding of the work, Collaborative construction

Table 1. Definition of Specification as a Complex Activity

4.2 Dialogue model

We had focused our study on the debate mode. Moderated debates are carried out through different stages related to a phase of the task model. There are two aspects, which become relevant: The structure and form of a debate and its relationship with the activity and domain model. For the specification task, the debate starts after a personal work phase, and consists of a set of sessions: the *initial* session is usually bounded to introduce personal solutions. This session just consists of one round. A round is accomplished when every member of the group has participated once at least. A session can be a set of rounds. A message from the moderator opens the discussion starting a new session, then students try to understand the others' solution proposals. A number of comments are made and new solutions are proposed out of the old ones. This is called a *diversification* session, because new themes can appear in the debate. The debate is eventually focused by a message from one of the participants, opening a *consolidation* session, and can be followed by a new diversification session, and so on, until an *agreement* session is caused by the moderator. As mentioned before, the debate is established by means of electronic messages produced by the students using a conferencing system. Messages related to a collaborative phase for constructing a solution can be roughly classified in two groups as follows:

1. Those referring to domain contents (IS, IRS, CS). Introduce solution (IS): this type of message is only sent at the beginning of the debate. Its goal is to share solutions with the other members of the group. Introduce reviewed solution (IRS): Normally, during a debate, one has to review its last introduced solution due to others students comments. The goal of this message is to propose a reviewed solution. Comments about solution (CS): comments can be of any type (suggestions, criticisms, questions, misunderstandings) and can refer to any of the solutions introduced by the members of the group.

2. Those concerned with the task and debate evolution (CTM, CC, R). Comments on task management (CTM) for instance to summarize the situation up to the moment. Comments on coordination (CC): They are mainly done by the moderator, they can refer to time scheduling, handling of participation. Reflection (R): a variety of written thoughts, expressing personal/ interpersonal dispositions.

Messages of the first group create new objects or links on the shared domain workspace, so that everyone can view for a proposal, new contributions as well as all the participants' comments on it. Providing a semi-structured way to perform interactions allows to create automatically links between the domain, activity and conversational models. This structured and interrelated information can be used in different ways, for instance by a teacher or the students to know the work progress, to focus on the current state of affairs, to analyze contributions to a particular proposal, or to further evaluate whether and how collaboration occurs. Furthermore, the system can have information, to some extent, about how the debate is going on. This, eventually, provides a potential source to be exploited for pedagogical support, as for example, to design and include a coaching facility to stimulate group interaction.

4.3 Domain Model

To support problem-solving in a domain, a variety of knowledge sources and tools can be foreseen. Sources may include declarative knowledge, for example the criteria to use a particular technique, and procedural knowledge i.e. how to proceed to solve a type of problem, with available examples from a case-based library. Tools like compilers and debuggers provide dynamic models to be used in some stages of the learning process.

Central to our application is a shared tool: a common workspace where students can write their solutions for a given problem using a graphic editor. Based on the entity-relationship model, the notation is precise enough as to be interpreted unambiguously, and powerful enough to describe the solution at different levels of abstraction. A semi-graphical representation has the advantage of being easy to use so that students can concentrate in the problem itself and not to bother about the features of a new formal language. An advantage of this type of representation is, that to some extent, an object can be automatically analyzed, and then the system can help in the comparison of the variety of solutions worked out, providing operations to answer questions such as the number of different techniques proposed by the group to solve a problem or check whether the restrictions for the input in a set of proposal are the same. The workspace model includes a set of predefined objects, built up from primitive elements, so that students can either built their objects from scratch (using the primitive operations) or by means of selecting and filling operations. The level of abstraction and grain size of the objects and the operations provided to manipulate them, is a tool design issue of great importance. Combining a graphical and textual representation for operations such as to visualize, edit, search and compare seems intuitively a sensible approach.

4.4 The learner interface

The learner interface aims to make easy creating and discussing a solution for a given problem. Figure 1 shows a screen shot from a *mutual understanding* activity.

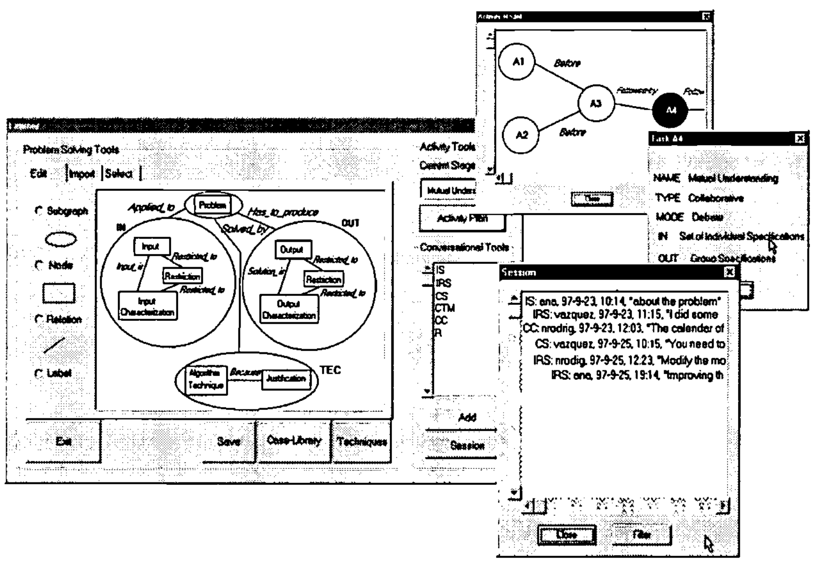

Figure 1. Learner Interface

The main window is divided in three areas: problem solving on the left, activity and conversational spaces on the right. The problem solving area includes a shared workspace were contributions can be written, either by selecting and/or editing a previous proposal, or by importing a new one. A set of buttons gives access to the case-based and technique libraries, where navigation is facilitated to find easily the information required. The discussion tool offers a set of typed messages. Students can discuss by posing comments to proposals, questions,... and responding to them. All the messages for the session can be visualized on the session window and the filter option provide a variety of views to see only kinds of interconnected messages. This semi-structured support facilitate learners to focus on the content of their discussion The activity button give access to the activity plan, to visualize the requirements of the current step, or to know about the next/past activities. This tool facilitates to students involved in the task, and also to teachers, to monitor their progress.

5. Conclusions

A number of experiences, involving students, have been undertaken with a previous version of the system prototype. The application described above, include all the changes derived from observation, analysis and discussion of the students activities. For the case study, we have used as domain subject a course on Algorithmics, run on the second year of a Technical University. Students following this course were involved in the design and implementation of an information system for the management of a library. They have previously done some groupwork and have positive feelings towards the use of electronic communication. The specification step was defined as the main focus of attention for analysis and assessing purposes. Scenes have been selected to study particular modeling aspects. The experiment was conducted with different groups, each one involving three students. The activity included an individual work phase followed by a collective debate to produce the collaborative solution. The concatenation of several phases covering the whole process has not been tested yet. Every group has been provided with the same problem. As these experiments went on, guidelines to incorporate pedagogical support on the models were established. As well, the mapping of learners activities on interface actions was clarified. Roles, particularly the moderator, were clearly seen as useful.

Our goal is to encourage distance learning students to work together, in order to promote both learning of collaboration and learning through collaboration. Collaboration means working together on a common problem, communicating and coordinating activities towards a shared goal. We propose a system embedding three models: an activity model, a domain model and a conversational model. Based on these models, a computer system can support asynchronous collaboration in a number of dimensions: giving structure to activities and communication to articulate the problem solving-task, supplying a share space to build jointly common knowledge, providing relevant know-how by case and techniques libraries and facilitating further reflection on the whole learning process. Experiences and preliminary results have convinced us that this approach offers promises to establish collaborative distance frameworks to enhance students learning processes.

References

[Collis 94] Collis B. (1994) Collaborative learning and CSCW: research perspectives for interworked educational environments. In *Lessons from learning.* R.Lewis, P.Mendelsohsn (Eds). North-Holland. 1994.

[CACM 96] Special Issue on Learner Centered Design. *CACM*, April 96, v. 39, n4.pp 32-33.

[De Cindio et al. 88] De Cindio, F., Simone C. Vassallo, R. Zanaboni, A. (1988) CHAOS: a knowledge based system for conversing inside office. In : W. Lamersdorf (ed). *IFIP tc8/WG 8.4 Int Workshop on Office Knowledge: representation, management and utilization.* North-Holland 1988.

[Edelson et al. 96] Edelson,D.C., Pea R.D. ,Gomez L.M.(1996) The collaboratory Notebook. *CACM*, April 96, vol 39, n4.pp 32-33.

[Gould & Lewis 87] Gould,J.D., Lewis, C. (1987) Designing for usability; Key principles and what designers think. In R.M.Baeker & Buxton (Eds) *Readings in human interaction.* San Mateo,CA: Morgan Kaufman. 1987.

[Malone & Fry 92] Malone, T.W., Fry,C. (1992) Experiments with OVAL: a radically tailorable tool for cooperative work. *Proceedings on CSCW-92* Toronto, November .

[Norman & Draper 86] Norman, D., Draper,S.W. (1986) User-centered system design. Hillsdale,N. J. Lawrence Erlbaum Associates.

[Verdejo 96] Verdejo, M.F. (1996) Interaction and collaboration in Distance Learning through computer mediated technologies In *Advanced Educational Technology: Research Issues and Future Potential.* T.Liao (Ed). *Computer and System Sciences*, vol 145, pp 77-88. Springer-Verlag.

[Wan & Johnson 94] Wan D., Johnson P. (1994) Experiences with CLARE: a computer-supported collaborative learning environment. *Int.J.Human-Computer Studies*, vol. 41, pp 851-859.

Akcnowledgements

This work has been partially funded by the Spanish *Ministerio de Educación y Cultura* through *Acción Integrada* HP97-0050 and by CICYT, project number TEL97-0328-C02-01.

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").