An Evolutionary Approach to Prototyping Pedagogical Agents: From Simulation to Integrated System

Anders I. Mørch, Jan A. Dolonen, Jan Eirik B. Nævdal

Prepublication version of "Mørch, A.I., Dolonen, A.I. and Nævdal, J.E. (2006). An Evolutionary Approach to Prototyping Pedagogical Agents: From Simulation to Integrated System. Journal of Network and Computer Applications, 29(2-3), 177-199."

An Evolutionary Approach to Prototyping Pedagogical Agents: From Simulation to Integrated System

Anders I. Mørch, Jan A. Dolonen, Jan Eirik B. Nævdal
InterMedia, University of Oslo
P.O. Box 1161 Blindern, N-0318 Oslo, Norway
E-mail: anders.morch@intermedia.uio.no

Abstract

We have developed and integrated software agents with two educational groupware systems (TeamWave Workplace and FLE), using evolutionary prototyping and empirical-based design as development techniques. The resulting prototypes of pedagogical agents (CoPAS, SA-Agent and RuleEditor) provide learners and teachers with increasingly domain-specific support for distributed collaborative learning activities. Employing the evolutionary approach has enabled us to build and evaluate early prototypes of complex systems with cost-effective techniques and involving users in this process helped us to constrain the design space and direct further development. CoPAS is a simulation experiment carried out with the Wizard-of-Oz technique, SA-Agent is a pedagogical agent integrated with an open-source learning environment, and RuleEditor is a customizer for the SA-Agent. The agents collect statistical information on user activity and analyze that information based on principles of collaboration and knowledge building. The results are presented as advice in the user interface of the learning environments to promote students' reflection on their collaboration and knowledge-building activities. If instructors disagree about the phrasing of the advice or the frequency of intervention, they can change it using the RuleEditor agent customizer.

INTRODUCTION AND MOTIVATION

Distributed collaborative learning is a new application domain for interactive learning environments, integrating Web-based communication and information sharing tools with reusable learning objects (multimedia resources originally made for CD-ROMs). The resulting systems are referred to as educational groupware (Gutwin, Stark & Greenberg, 1995; Toth, Suthers & Weiner, 1997), E-learning (Rosenberg, 2001), or LMS (Learner Management Systems) depending on whether the focus is on tools, learning objects, or technology infrastructure, respectively. Examples are TeamWave Workplace (Roseman & Greenberg, 1996), Belvedere (Toth, Suthers & Weiner, 1997), FLE (Fle3, 2004), WebCT (WebCT, 2004) and ClassFronter (Fronter, 2004). The impact of this technology on education is expected to be substantial, and it has attracted interest from a broad range of stakeholders, including educators, researchers and developers. With educational groupware students are empowered to interact with fellow students in new ways, allowing them to accomplish learning tasks with more flexibility. At the same time, teachers can monitor participation and initiate explorative learning activities. In many cases, students' assigned tasks transcend the physical constraints of the classroom; settings outside of school also work, so long as the resources required for learning are available.

Knowledge building (Scardamalia & Bereiter; 1994) and its various adaptations (Stahl, 2000; Hakkarainen, Lipponen & Järvelä, 2002; Law, 2002; Wasson & Ludvigsen, 2003) is a pedagogical model that fits well within this framework of teaching and learning. In its basic form, knowledge building is about raising questions that trigger prolonged discussions. More formally, such questions are followed by alternative answers (hypotheses or student explanations) backed-up by arguments that reference scientific knowledge. This pedagogical model allows for the progression of student inquiry from a vague problem or poorly formulated question toward a clarification of the phenomenon being studied. In a good learning process, this will lead to scientific explanations that can be shared by a community of learners in which the participants are either co-located and/or geographically distributed.

The teacher's role in this sort of environment is more that of facilitator than that of someone who stands in front of the class and "preaches". In this context, it is believed that knowledge-based software agents can take on the role of a "substitute facilitator" in place of the teachers when it is difficult or impossible for them to participate (for example, after working hours or during remote collaboration sessions). This is possible even though the software agents do not have a detailed model of the knowledge domain, do not know the background of the students, and do not simulate human advice-giving with lifelike behaviour. Although this has been featured in previous research (e.g. Johnson, Rickel & Lester, 2000), Constantino-Gonzalez and Suthers (2001) have found that reasonable collaboration advice from a virtual coach could be generated without the need for expert solutions or discourse understanding. Furthermore, Craig, Gholson & Driscoll (2002) have found that although lifelike characters are perceived as useful, there is no indication that they lead to increased learning performance when compared to other forms of feedback containing the same information. A low-cost alternative is letting interface agents operate on the shared state of a groupware and collect user interaction data, which is then returned to the students in ways that are meaningful to them. For example, agents can observe who is logged on, who communicates with others, what shared objects they act upon, how much of a given task has been completed, etc. When agents are allowed to reason with this information, they can assist students with collaborative learning tasks. In this paper we address this within the context of two existing educational groupware systems, TeamWave Workplace and FLE (Future Learning Environment).

The paper is organized as follows. We begin by presenting the questions that has motivated our research. Next, we survey previous work in CSCW, CSCL and interface agents. We then present a series of prototypes we have developed, each of which provides increasingly specialized support for distributed collaborative learning, starting with generic collaboration and ending with knowledge building and adaptability. We conclude by comparing our approach to related work in these areas.

RESEARCH QUESTIONS

Computer-supported collaboration and knowledge building are complex activities that are only partly supported by the current generation of knowledge-building environments. One reason for this is that collaboration tools such as discussion forums and chat rooms are mostly *generic* (domain-independent), thus requiring students to familiarise themselves with new modes of learning that are different from traditional classroom activity. Furthermore, intensive collaboration may include a large number of participants, who are geographically separated, have a high number of relevant postings, or a multitude of simultaneous activities in a real-time program, making participation, communication and coordination demanding. Although postings in a discussion forum may be organized around shared conventions such as topic threads, text indentation and graphical symbols; and while concurrency control will allow multiple users to interact simultaneously in group editors, it still takes time and effort for someone to analyse and coordinate the overall activity, detect potential communication and collaboration problems and follow up with useful advice in a timely manner. Both teachers and computer-based tools (e.g. agents, awareness mechanisms) should be engaged in this activity because there is plenty of opportunity for scaffolding in these environments (Soller et al., 1999).

Based on our preliminary observations, we have developed the following working hypothesis: on the one hand there are theory-based models of collaboration and knowledge-building underpinning collaborative learning environments (e.g. Scardamalia & Bereiter, 1994; Soller et al., 1999; Hakkarainen, Lipponen & Järvelä, 2002); on the other hand, moment-to-moment student interaction with these environments is not model-driven, but is instead deeply contextualized and determined by the constraints and opportunities of situated actions (Suchman, 1987), including whom the students prefer to collaborate with, the subject being taught and the resources available in the course. These two sides rarely coincide, forcing students to appropriate the pedagogical tools in different ways (Arnseth et al., 2001). Many students find that model-driven approaches (e.g. sentence openers) are not very useful (Lazonder et al., 2003), while the complexity of some pedagogical models, such as progressive inquiry, put high demands on both students and teachers (Muukkonen, Hakkarainen & Lakkala, 1999). For example, with the FLE (Fle3, 2004) we have observed that students sometimes use only a subset of the reply categories they have at their disposal (such as Problem and My Explanation), even though they post scientific explanations (Ludvigsen & Mørch, 2003). As a result, the pedagogical principles behind the learning environment may not be properly communicated, resulting in students becoming frustrated and using the tools for purposes other than learning (Hara & Kling, 1999). How these two concerns (pedagogical models and situated activity) might be reconciled within the context of computer-based learning environments is our primary research question, and it can be broken down into the following two sub-questions:

- How can a distributed learning environment stimulate the transition from situated activity to collaborative learning?
- How can a distributed learning environment that has been built according to a generic pedagogical model be adapted to the needs of moment-to-moment student activity?

These two questions will be explored by software prototypes in the second half of the paper. In the next two sections they serve as lens with which to focus the discussion.

BACKGROUND

The general research areas of this paper are Human-Computer Interaction (HCI) and Knowledge-Based Systems (KBS), but more specifically Computer Supported-Collaborative Learning (CSCL), Computer-Supported Cooperative Work (CSCW), and Interface Agents (IA). The related work we survey is grouped into theories (CSCW, CSCL, knowledge building) and technologies (educational groupware, workplace awareness, pedagogical interface agents).

Computer-Supported Cooperative Work

Computer-Supported Cooperative Work (CSCW) emerged as an established research area during the mid-1980s by bringing together researchers from a variety of fields, including computer science (Ellis, Gibbs & Rein, 1991), information systems (DeSanctis, 1987), Computer-Mediated Communication (Hiltz & Turoff, 1985), HCI (Bannon, 1986), and ethnography (Hughes, Randall & Shapiro, 1992). A shared interest has been how collaboration technology impacts work environments and how this technology can best be developed (Grudin, 1994). The technology in CSCW is often referred to as groupware (Johnson-Lenz & Johnson-Lenz, 1982), but it encompasses a wide range of systems, including e-mail, news groups and bulletin boards, information sharing systems (such as discussion forums) and advanced applications for facilitating synchronous communication such as group editors, multimedia spaces and video conferencing systems. During the past few years, the functionality of this technology has gradually matured and stabilized, with the results that many of today's desktop computers provide built-in groupware tools catering to a host of cooperative work arrangements, and that there are emerging standards for information sharing, concurrency control and awareness technology.

Computer-Supported Collaborative Learning

Computer-Supported Collaborative Learning (CSCL) is a new field that emerged during the last decade and concerns educational technology and its role as mediator of activity within a collaborative setting of instruction and learning (e.g. Dillenbourg, Baker, Blaye & O'Malley, 1996; Koschmann, 1996). Groupware is often used as the technological platform for achieving this. CSCL as a field is complex and multifaceted, borrowing ideas from such social sciences as sociology, anthropology, linguistics and communication (Koschmann, 1996). Researchers and educators in these fields have questioned and rejected earlier approaches to computer-based instruction because learning in those fields is perceived as belonging primarily to the realm of cognitive psychology. The focus in CSCL has instead shifted to socially constructed properties of interaction (Dillenbourg, et al., 1996). From this perspective, knowledge is viewed as a human construction elaborated through communication and collaboration with peers and mediated by social and cultural artifacts (e.g. language, technology), implying that learning and knowledge building occur first on interpersonal grounds within a community of learners and more experienced peers and facilitators before they occur in the intrapersonal realm of an individual learner (Vygotsky, 1978).

Knowledge Building

A pedagogical model developed within this perspective is knowledge building (Scardamalia & Bereiter, 1994). Knowledge building assumes that new knowledge is not simply assimilated with the help of a more knowledgeable person, but that it is also jointly constructed by problem solving with peers through a process of building shared understanding. Knowledge building and its subsequent refinement, *progressive inquiry*

(Hakkarainen, Lipponen, & Järvelä 2002), have received considerable attention in the CSCL community because it fits well with the educational philosophies of many schools in Canada and Scandinavia (problem-based learning) and elsewhere in the world. The basic idea is that students will gain a deeper understanding of a knowledge domain if they engage in a research-like process in the domain, generating their own problems, proposing tentative hypotheses and searching for deepening knowledge collaboratively.

Systematic (well-defined) domains have shared principles and rules of behavior acknowledged by experts as useful for others to know in order to successfully navigate within the space of potential moves. Although these principles and rules are harder to identify in less stringently structured domains such as collaborative learning, researchers in CSCW and CSCL have identified recurring relationships that need to be satisfied before a shared group task can successfully be completed. We call these relationships *collaboration patterns* (Wasson & Mørch, 2000), and they include common goals, shared feedback (Dourish & Bellotti, 1992), mutual learning (Bratteteig, 1997), sharing information, division of labor and thinking jointly in explicit terms (Salomon, 1992).

In the same way, the regularity of knowledge building and progressive inquiry defines a systematic domain, namely *scientific inquiry*. This is manifest each time a message is posted by a student in the discussion forum of knowledge-building environments such as FLE, because the student must select a reply category reflecting the stage of the inquiry process in which they are involved at the time they post the message (Muukkonen, Hakkarainen & Lakkala, 1999). Scaffolding knowledge building includes: 1) helping students to choose which category to use when proposing a '*problem*', 2) teaching students what a '*my explanation*' is, 3) helping students use personal explanations in relation to a problem or a '*scientific explanation*' to take the inquiry to a general level, 4) finding deepening knowledge to back up explanations, and 5) collaborating with others during these processes (Ludvigsen & Mørch, 2003).

Educational Groupware

The two groupware systems we have worked with in empirical studies and system-building efforts are TeamWave Workplace and Future Learning Environment.

TeamWave Workplace (TW) (Roseman & Greenberg, 1996) is a groupware system developed at the University of Calgary, Canada that provides a set of communication and information sharing tools that allow multiple users to work on shared objects simultaneously, such as editing diagrams or concept maps. The system supports workplace awareness, providing up-to-the-minute information about the state of others' interactions with the environment, which can help to reduce the work of coordinating common tasks and actions relative to shared objects (Gutwin, Stark & Greenberg, 1995). TW is based on the room metaphor, providing the tools with which to "furnish" (configure) virtual places or interactive areas inside the environment. These rooms can be configured independently, according to one's needs. The tools used in the CoPAS experiment (see below) were Concept map, Chat room, and Post-it notes. Figure 1 shows the initial set-up of the environment for this experiment.

Insert Figure 1 about here: TW tools used in the CoPAS experiment

FLE (Future Learning Environment) is an open-source learning environment developed at the University of Art and Design in Helsinki, Finland. FLE is an asynchronous, web-based groupware system for computer-supported collaborative learning (Fle3, 2004), designed to support collaborative learning in the form of discussion forums with message categories (knowledge types) named after the stages of the progressive inquiry model (Muukkonen, Hakkarainen & Lakkala, 1999). FLE users are required to select one of these categories each time they post a message in the forum. Figure 2 shows the writer and reader interfaces of the knowledge-building module of FLE. In the version we adopted, the type set was first translated into Norwegian. The Norwegian version is slightly different from FLE, since we split one of the categories (Deepening Knowledge) into two other categories: Reliable Knowledge and Uncertain Knowledge.

Insert Figure 2 about here: Future Learning Environment.

Workplace Awareness

Awareness is a central feature of educational groupware and a technique used to help students coordinate common actions and activities relative to shared objects. It is most prominent in synchronous (real-time) environments because the need for conveying up-to-the-minute information about the common actions and activities of collaborating peers is most present there, aided by techniques such as multiple cursors and echoes (Beaudouin-Lafon & Karsenty, 1992) and knowledge of who is looking at or working on shared objects (Ogata, Matsuura & Yano, 2001). Such techniques can help to convey a single user's action on the shared environment to the others, but awareness is also common in asynchronous collaboration environments such as interaction histories, structuring of discussion threads and electronic Post-It notes (Gutwin, Stark & Greenberg, 1995). A common definition of awareness describes it as "an understanding of the activities of others, which provides a context for your own activity" (Dourish & Bellotti, 1992). The context can help users to interpret others' actions and ensure that individual contributions are relevant to the group's overall activity, and it enables evaluation of individual actions with respect to the group's goals. Gutwin at al. (1995) have suggested four types of awareness that may be relevant for this type of coordination support: workspace, social, task and concept. All of these are critical to the quality of collaboration and coordination in a CSCL environment, but it is workspace awareness that has received the most attention in previous studies.

Pedagogical Interface Agents

A pedagogical agent is an interface agent, a KBS term describing a semi-intelligent system that assists users with routine tasks such as searching for information in large information spaces, or monitoring actions on computer screens in order to give advice to users in complex problem domains (Maes, 1994). The most common application of interface agents is integration with existing systems. A range of agent-enabled applications exists, including email (Malone et al., 1987), graphic editors (Lieberman, 1998), design environments (Fischer, McCall & Mørch, 1989), spreadsheets (Krogsæter, Oppermann & Thomas, 1994) and WWW applications (Terween & Murray, 1996; Mørch, Dolonen & Omdahl, 2003).

We have adopted a notion of pedagogical agents originally proposed by Johnson, Rickel and Lester (2000), who defined them as autonomous and/or interface agents that support human learning by interacting with students in the context of an interactive learning environment. The context for their work has been the Intelligent Tutoring Systems (ITS) framework (Johnson & Rickel, 1998). Our notion of pedagogical agents thus differs from past work in that we focus on how interface agents can support collaborative learning, and consequently how they can be embedded in CSCL environments. From our perspective there is convergence between the computational processes expected of a pedagogical agent and the task and concept awareness mechanisms for educational groupware proposed by Gutwin, Stark, and Greenberg (1995). This is possible even though the agents do not have a detailed model of the knowledge domain (Constantino-Gonzalez & Suthers, 2001) or a presentation style simulating human body language (Craig, Gholson & Driscoll, 2002). By operating on the shared state of a groupware environment, the agents can observe who is logged on, who communicates with others, what objects they act upon and how much of a given task has been completed (McManus & Aiken, 1995). Agents can organise and present this information so that students are guided in their collaborative learning efforts (Mørch, Dolonen & Omdahl, 2003) and teachers are provided with tools for monitoring and coordinating the overall activity (Chen & Wasson, 2002).

To achieve a synergistic effect with agents in learning environments they must make their presence known at the right time. If they fail to do so, they may be ignored or perceived as disturbing ongoing activities. Nardi (1993) predicts that the best agents are highly malleable, programmable tools which empower rather than diminish users' roles and give them improved control over the tasks they are required to accomplish with computer systems. The work on programmable agents that is most applicable to our own efforts is the work of Fischer and colleagues at the University of Colorado (Fischer, McCall & Mørch, 1989; Fischer & Girgensohn, 1990) who have integrated agents (referred to as critics) with design environments, casting them as advice givers and design support systems. The rationale for this approach is rooted in the contradiction that, on the one hand, design is the exploration of alternatives within a space of possible moves; while on the other, it is bounded by a set of constraints represented by guidelines and rules. This is analogous to the contradiction identified above for educational groupware (pedagogical models vs. situated activity). Customization support is needed in these settings because design rules, such as building codes and designers' preferences, change over time and new design units are periodically introduced (Fischer & Girgensohn, 1990).

APROACH

We have employed an evolutionary systems building approach. Each prototype created after the first reuses ideas or components from its predecessors, but not in its entirety. Functionality that was unsuccessful in previous versions has been removed. When we began our work in the spring of 2000, we did not have a precise idea of what we wanted to achieve. We therefore started with a simulation experiment known as a "Wizard of Oz study". This cost-effective technique is often used test out new ideas before they are implemented: participants are led to believe they are interacting with a software system, when in fact they are interacting with a simulation staged by human experts (Dahlbäck, et al., 1993). When using state-of-the-art collaboration technology, the cost of implementing new functionality (such as pedagogical agents) can be high, especially when the technology is not built for adaptability. Using the Wizard of Oz technique we have been able to create a realistic simulation of a knowledge-based, collaborative learning environment in a shorter time than it would have taken were we to have programmed the agents in a programming language. The method has allowed us to try out functionality that we later refined in software prototypes. In a similar cost-effective manner, one of the prototypes has been integrated with an open-source learning environment, allowing us to peek into the source code of the system to see exactly what was going on and enabling us to find the variable and methods we needed access to more easily. This was a considerable help to us when we integrated the SA-Agent with FLE. One disadvantage of this approach is that it takes time to read and understand existing source code. In our case, the advantages outweighed the disadvantages because the code was well structured and easy to read.

SIMULATION EXPERIMENT

The Wizard of Oz experiment was conducted in the TeamWave Workplace (TW) environment (see Figure 1). Participants were given a short introduction to TW before the experiment started, including how to use the different tools and how to query the domain agent for advice. Six groups of freshman students from the Information Science Department at the University of Bergen participated. The students were organized into groups of three, with each group given the task of creating object-oriented analysis and design (OOA/D) diagrams for an Internet banking system during a 90-minute session. The fixed time duration was judged appropriate in this case; not only because the assignment was taken from a previous course exam, but also because the wizards (human agents) might have revealed themselves if they were active for a longer period of time, since it would increase the chances that the students guessed they were part of an agent simulation study.

Three graduate students acted as wizards, simulating the following agent functionality: 1) Tool Agent, 2) Domain Agent and 3) Collaboration Agent. The tool agent had technical knowledge about how to use the TW tools, the domain agent provided assistance on object-oriented analysis and design, and the collaboration agent provided assistance on how to interact based on collaboration patterns (Salomon, 1992; Wasson & Mørch, 2000). The wizards were given oral instructions for how and when to act and had a list of rules for what to say upon acting. Two of the agents (Tool agent and Collaboration agent) were active agents, meaning that they took initiative, responding to the situation whenever they saw an opportunity. The domain agent was a passive agent, taking action only when it was requested. The presentation technique employed was text messages displayed in pop-up dialogs, a technology readily available in TW as a command in its designer substrate (Roseman & Greenberg, 1996). An example of an agent's message is shown in Figure 3. It is a "message to all users," which is a technique for sending shared feedback in TW.

Insert Figure 3 about here: A pop-up box from the Collaboration agent.

As a passive agent, the Domain agent takes action only when a user requests it. Figure 4 illustrates a request for help from this agent. Clicking on the Domain agent icon allows the user to page the agent, bringing up a dialog box where the user can type in a question. This question is then sent to the wizard, which provides a response.

Insert Figure 4 about here: Steps for querying the Domain agent.

The Domain agent responded by sending feedback only to the user who requested the advice and not to all users in the environment. This was accomplished using the same technique as when the agent was questioned in the first place: a TW "page," or private message sent to a single user. When none of the predefined messages applied, the Domain agent either made up a new message to salvage the situation or responded with the following message: "Domain agent cannot understand your request. Please reformulate your question." The wizard issued this message quite often since many questions were unanticipated and therefore difficult to answer on the spot.

Table 1 shows the total number of messages sent by the wizards to the six groups. Of the original 38 message alternatives they could choose from, 25 (65.5%) were used at least once. The overall picture is that there is a relatively good match between the type of messages prepared in advance and the messages actually used. However, the wizards created a total of 19 new messages during the course of the experiment. They were neither told to do so nor prevented from doing so. During the interviews afterward, the wizards told us that they had to make up these messages when the students were stuck in difficult situations for which there were no predefined messages to assist them.

Insert Table 1 about here.

The CoPAS study (Jondahl & Mørch, 2001) helped us to identify the following areas for further work on integrating pedagogical agents with collaborative learning environments (the list is not exhaustive, but identifies the problems we chose to work on):

- The collaboration agent should be refined (from general to specific collaboration support).
- Active agents should replace passive agents in order to support shared feedback (feedback to all users).
- Pedagogical agents should be customizable when the task domain is poorly understood.

We address the first two issues in the next section (SA-Agent) and the second issue in the subsequent section (RuleEditor).

STUDENT ASSISTANT (SA) AGENT

In the second agent prototype (Dolonen, Chen & Mørch, 2003) we shifted the focus from general to specific collaboration support, and the educational groupware from TW to Future Learning Environment (FLE) (see Figure 2). A key feature of FLE is that knowledge-building categories have been implemented in the discussion forum as a scaffolding structure that guides the progressive inquiry (PI) process (Muukkonen, Hakkarainen & Lakkala, 1999), requiring students to select a reply category (knowledge type) that match the PI model each time they post a message. Table 2 compares the knowledge types of Fle2 and Fle3. Both versions were used in our trial studies because FLE was upgraded from version 2 to 3 before we completed our system building efforts.

Insert Table 2 about here: Knowledge-building categories in FLE

The design of the Student Assistant Agent (SA-Agent) was guided by a pilot study in which we tested the use of FLE without using agent assistance (Mørch, Dolonen & Omdahl, 2003). Two 9th grade classes from two schools in Norway (students and teachers) took part in the study. The students had never met each other before. After an introduction to the progressive inquiry model and FLE tools, students from one school were asked to engage in a knowledge-building activity with students from the other school (each class was divided into seven groups of 3-4 students that were paired across the two schools). The activity lasted one hour a day for two weeks. Natural science was chosen as the subject domain because it fits well with the problem-based and collaborative learning philosophies of encouraging science discussion in the schools. Controversies in genetic engineering served as triggers for knowledge building, because we hoped that this topic would be personally meaningful to the students. To stimulate their interest, we presented a 15-minute video on biotechnology that had previously been broadcast on national TV. After viewing it, the students formulated such questions as whether an animal heart can replace a human heart inside a human body; whether a Muslim can receive a pig's body part during a transplant; and whether it is dangerous to eat genetically modified food. These questions were later entered into FLE to initiate a knowledge-building session with students from the other school.

Although the initial questions were well formulated and easily entered into FLE, responding to them by classifying the answer by knowledge type was more difficult. The students had particular difficulty distinguishing between My Explanation, Summary, Meta-comment, and Scientific Explanation. On several occasions, students chose the "wrong" category for a posting, as can be seen in the following example (message posted by one of the groups during the beginning of the session): "Before we can begin we need to decide whether we should be for or against genetically modified food?" This message was posted as a My Explanation note rather than a Meta-comment, illustrating the difficulty of choosing proper posting categories, and especially the uncertainty of choosing between these two categories (Ludvigsen & Mørch, 2003).

After the two-week pilot study trial, we coded the postings by category/content matching to see whether the students were in fact employing knowledge building, or if they were performing other activities, such as socializing (i.e. arbitrary choosing reply category). We selected the following three discourse types as the coding scheme: knowledge-building proper (KB), meta-commenting (MC), and social talk (ST). We identified the content (message body) of each note in the database as belonging to one of the three types. When two or more discourses were present in the same posting, we coded it as KB (if present) or MC (if present). Cases with no clear match were coded as ST. The coding scheme is a variation of a scheme originally proposed by Svensson (2002), who identified Query, Feedback and Smalltalk as three types of postings that commonly occur in web-based discussion forums. Svensson called these discourse types *genres* and noted that they capture both formal and informal information exchange (Svensson, 2002). We observed a similar behaviour in our study.

A summary of our findings (Mørch, Dolonen & Omdahl, 2003) shows that the students were primarily employing knowledge building (62%), but meta-commenting (25%) and social talk (17%) occurred as well. This was not surprising, since it was the students' first encounter with knowledge building, but we did not expect the relatively high number of meta-comments. We found this to be a double-edged sword: on the one hand, meta-communication is a desired outcome of knowledge building when it helps to advance a student's knowledge level from mastering basic skills (e.g. the basic steps of progressive inquiry) to higher order skills. Meta-commenting is considered a higher order skill in knowledge building when it prompts students to reflect on the learning objectives, restructure their learning tasks, etc. When meta-commenting was attempted without prior mastery of basic skills it often led to unproductive interaction and social talk, possibly because the meaning of the meta-commenting category is difficult to comprehend, thus encouraging alternative (shortcut) interpretations. When meta-commenting shortcuts into uncompleted arguments or social talk, it is called 'premature meta-commenting' (Fugelli, 2004). A major part of the meta-commenting posts in our study were of this sort. This identifies an area for further research on collaborative knowledge building.

Based on the above findings, we have implemented the SA-Agent to guide the student knowledge-building process (Dolonen, Chen & Mørch, 2003). This agent has been integrated with FLE to advise students on how to choose the best knowledge-building category for a new posting, based on postings that have already been submitted. The agent is implemented in Java, but we have not made any changes to the functionality of

FLE (written in Python). The role of the agent is twofold: 1) to monitor the knowledge-building forum, and 2) to present advice (generalized feedback) to the students.

When monitoring the forum, the SA-Agent gathers statistical information about user activity and stores it in a database. For example, how many notes has a certain student posted? How often does a certain student post notes? How many notes have been posted in each category? How many notes has each student posted in a certain category? The stored information includes the structural properties of the messages, which is processed by a rule-based inference engine. In here trigger values are computed and relevant rules are submitted to a firing agenda where they are organised according to importance factor. Finally, an advice generation module selects the rules and presents feedback to students in the user interface of the web browser.

The SA-Agent presents its advice in a dedicated window in the FLE discussion forum, or, alternatively, as a pop-up window superimposed on the forum. Examples of messages sent by individual agents are:

1. "You have posted many more messages than the others. Make sure you do not dominate the discussion and prevent others from participating";
2. "Several notes have been posted since you were last logged in. Please make an effort to answer some of them";
3. "There are many "Problem" postings in the thread. Although a "Problem" can be followed by a sub-problem, you should try to respond using "My Explanation";
4. "There is a "My Explanation" note without any response. You should read that note and try to respond using a "Comment" or "Scientific Explanation".

The first type of advice is generated when the corresponding rule's "measure of participation" exceeds a predefined threshold value. A student's postings are counted to compute this measure. The second type of advice is computed based on a count of the number of postings that have been submitted by others while the student has been logged off. The third applies when there are more than three consecutive “Problem” notes in the same thread. Similar rules have been defined for the other knowledge-building categories. The last example reminds the users that there is a note awaiting a response. When a computed value exceeds a predefined threshold value (trigger count), the corresponding advice is sent to the FLE user interface and presented in a separate window. An example is shown in Figure 5.

Insert Figure 5 about here: Student Assistant Agent

RULE-EDITOR: THE SA-AGENT CUSTOMIZER

Failure to provide proper assistance in an agent-enabled learning environment could result in messages being ignored or, even worse, being inhibitive to productive interaction. This dilemma is especially pertinent in poorly-structured knowledge domains because chances for an agent to ‘miss the point’ are much greater there than they are in well-defined domains. For example, during the CoPAS experiment, the agents (wizards) made up new rules when the students were stuck in difficult situations. A total of 19 out of 59 rules (33%) had to be created on the fly during the session. The SA-Agent also operates in an ill-defined domain. One of its limitations is that its rules are programmed in Java; with threshold values (trigger counts) that are hard-coded and feedback messages that are fixed (non-context sensitive). This makes them “brittle” and difficult for end-users to modify.

To cope with these shortcomings, we have built an rule-based agent customizer (Mørch & Nævdal, 2004). This customizer, called RuleEditor, is integrated with the SA-Agent and supports the editing and creation of rules by employing a high-level, easy-to-use tailoring language similar to the simplified query language used in most search engines. The customizer makes use of the same variables and methods as the SA-Agent, but does not make any changes to them; it is a separate component that can be independently deployed.

The usability of the customizer can be assessed by how well it bridges the "cognitive distance" that exists between use mode and tailoring mode (Malone, Lai & Fry, 1995; Mørch, 1997). This distance must be easy to travel for a system to be easy to modify, thus yielding high customization usability (Mørch, 1997). Rule editing with the RuleEditor is activated in the FLE's user interface through the SA-Agent's presentation mechanism (Figure 6). When a user chooses to edit a rule, he or she signals this by invoking an "*Edit Rule*" command with a single mouse click. A wizard then guides the user through the task of modifying the various aspects of a rule (trigger values, level of importance, display message), as shown in the bottom window of Figure 6. This results in a new display message, firing interval and importance level for the agent. The ease by which the agent can be modified provides a remedy for its perceived brittleness, i.e. direct manipulation of attribute values rather than editing Java code in a text editor. These values can be fine-tuned until the rule functions as desired by the user, because once modified, the SA-Agent is immediately re-evaluated.

Insert Figure 6 about here: RuleEditor interface

Rule generation is very similar to rule editing. The user goes through the same sequence of wizard screens, but this time they are filled out from scratch and the customizer must be invoked by a separate command *("New Rule").* When a new rule has been saved, it is automatically named and included in the rule interpretation loop according to its priority (level of importance) and type (what variables it applies to). An old rule can be removed from memory when it is no longer needed, but this is currently accomplished by lowering the rule's trigger values, not by physically removing it from the program's runtime image. This is to prevent users from deleting rules by mistake.

The RuleEditor is programmed in Java as an application and communicates with the SA-Agent and the FLE database, from which it receives all the information it needs. Agent rules and database variables are represented in a high-level rule-based expression interpreter that runs in the Java virtual machine. This 'language' is remarkably simple, like a query language used in most search engines, because it is well defined and task-specific from the point of view of end-user development (Nardi, 1993; Terween & Murray, 1996). A customizer rule has the following generic structure:

```
IF <database variable> [<absolute value> OR <average>] IS <trigger
value> [constant OR range] AND (NOT <inverse trigger value> [constant
OR range]) AND <rule trigger value> THEN <text message to be
displayed>
```

The above rule reads as follows with the values shown in Figure 6:

```
IF database variable Unreplied_notes average IS greater than 5 AND
rule trigger value is Less_important THEN text message to be is
“Several notes have been posted since you were last logged in. Please
make an effort to answer some of them”.
```

This rule is triggered when there are more than five unanswered messages in the discussion forum, but the rule has a low importance level, which means it will not repeat itself if the user does not take any action after it is presented . Figure 6 shows an example of how this rule can be presented in the FLE interface and how it can be customized in the RuleEditor. It is much easier to edit a rule in this way than it is to modify variables and methods in the corresponding Java program.

DISCUSSION AND COMPARISON TO RELATED WORK

The research questions we set out to address were 1) how a distributed learning environment can stimulate the transition from situated activity to collaborative learning, and 2) how a distributed learning environment that has been built according to a generic pedagogical model can be adapted to the needs of moment-to-moment student activity.

We addressed the first question by integrating pedagogical agents with two collaborative learning environments. These agents are able to detect situations when the students fail to engage in collaborative learning. This is based on the assumption that collaborative learning can be defined by a set of rules for how to collaborate. These rules are programmed on the computer and represented in pedagogical agents. When the students violate one or more of these rules the corresponding agents will trigger and the students will receive feedback, organized according to level of importance. The students can choose to follow the advice, ignore the feedback if they find it misleading, or they can use the RuleEditor customizer to “turn off” (lower the importance) of individual agents. Teachers can use the RuleEditor to fine-tune feedback messages and intervention intervals, so that offending messages are appropriately handled and important messages do not get unnoticed.

Adaptable or user tailorable agents address the second question. The lack of “hard rules” for defining how to interact in CSCL environments led us to a “soft rules” approach, meaning that the rules are flexible and can be modified at runtime (e.g. when a teacher detects that an agent has improper wording, or fires too often). The domain of collaborative learning may be better understood in the future, or specialized sub-domains may emerge. This may allow for a differential strategy for employing agents, supporting both “hard rules” and “soft rules”. The former may be associated with agents that are tightly integrated with other tools in the environment, like awareness mechanisms are realized in many groupware systems today. This identifies an area for future work on integrating agents with CSCL environments.

Alternative approaches to the same research questions include those of Belvedere and COLER (Toth, Suthers & Weiner, 1997; Constantino-Gonzalez & Suthers, 2001), the EPSILON project (Soller et al., 1999), Sharlok (Ogata, Matsuura & Yano, 2001), and Instructional Assistant Agent (Chen & Wasson, 2002), summarized below.

The work by Suthers and colleagues parallels our own efforts, but arises from a different tradition (Pittsburgh rather than Boulder). Belvedere is a collaboration system that supports students in the improvement of their critical thinking skills, and COLER is a web-based groupware system that supports learning Entity Relationship (ER) modeling. Each system has a coach who guides the process of building an artifact within a well-defined domain (“inquiry diagram” and ER diagram, respectively). The coach has knowledge about partial solutions and the relationships of elements in the diagrams, and can advise the students when certain elements should be rearranged to improve the corresponding artifacts. The CoPAS experiment and SA-agent borrow ideas from

Belvedere, particularly the collaborative aspects of coaching and the system architecture used for it. The main difference between the two approaches is that we focus on students who are geographically separated and provide users with tailoring tools to modify the agents.

Encouraging Positive Social Interaction while Learning ON-Line (EPSILON) (Soller, et al., 1999) is a related project that has investigated the integration of intelligent facilitation agents with a shared workspace of object-oriented analyses and design (OOA/D). These agents can observe a group's conversation and dynamically analyse individual student contributions. The dialog among students is scaffolded using sentence-openers modelled on speech act theory (e.g. justify, assert, encourage). The EPSILON agents are able to recognize events, such as a student completing a critical portion of the task, or a student failing to discuss his or her actions with others. When it detects an opportunity to react, the agent may intervene by asking the group to explain the student's actions. Both CoPAS and SA-Agent adopt strategies from EPSILON: CoPAS makes use of a similar task domain (OOA/D), and the SA-Agent makes use of computer-supported scaffolding resembling sentence-openers (FLE knowledge-building categories). A difference is that the SA-Agent operates in an asynchronous environment.

Ogata, Matsuura and Yano (2001) propose "knowledge awareness" as a concept for open ended CSCL environments. Knowledge awareness is workplace awareness (Gutwin, Stark & Greenberg, 1995) for collaborative learning. A goal has been to bring workplace awareness to a higher level, from the details of individual student actions and activities toward an awareness of the state of shared objects that has been collaboratively produced by a community of learners. Ogata, Matsuura and Yano (2001) have implemented this concept in the Sharlok system, a groupware supplemented with active and passive awareness mechanisms. These "agents" can inform a student that she is looking at the same object as another student and thus bring more attention to the shared object. Although originally developed for a synchronous learning environment, this type of awareness can be useful for asynchronous environments as well (e.g. postings in a discussion forum). We have borrowed the term knowledge awareness, calling it "conceptual awareness", to describe the SA-Agent.

The Instructional Assistant (IA) Agent (Chen & Wasson, 2002) is another agent integrated with FLE. The IA-agent has two roles: 1) to observe the distributed collaborative learning process and compute statistical information for viewing, and 2) to detect possible problems in the interaction and present them to the instructor so that the instructor, if desired, can give feedback to the students. It employs the same system architecture as the SA-Agent. The reason for including the instructor in the loop is to avoid situations in which the agent's understanding of the collaboration process precedes human judgment. This may lead to misinterpretation or misunderstanding among the students. Instructors can review the information before it is sent to the students and thus circumvent this problem.

SUMMARY AND CONCLUSIONS

In this paper, we have described a series of efforts to integrate pedagogical agents with two collaborative learning environments, one synchronous (TeamWave Workplace) and the other asynchronous (Future Learning Environment). We have also explored various ways in which pedagogical agents can be useful for both students and teachers within these environments; particularly in terms of facilitating collaboration and knowledge building but also encouraging active participation by all parties involved. Our system-building efforts have reused and extended previous efforts, both our own and those of others. Empirical studies have provided us with feedback to direct our work. Our agents collect information based on user activity in the environments and can 1) measure degree of participation, 2) encourage inactive students to be more active, 3) suggest what messages in a discussion forum should be replied to and by whom, 4) suggest what category should be chosen for the next posting during knowledge building, and 5) suggest when messages do not follow the recommended knowledge-building steps. If teachers disagree about the phrasing of the agents' advice or the frequency of intervention they can change it using the RuleEditor agent customizer.

Our results can be summarized as 1) a series of attempts that show it is possible to take advantage of statistical information in collaborative learning environments in order to scaffold aspects of collaboration and knowledge building, 2) categories taken from expert performance (scientific inquiry) that are useful for software agent scaffolding in a poorly-structured knowledge domain (science discussion in schools), and 3) customizable agents to address the imprecision dilemma associated with providing agent-based assistance in poorly-structured knowledge domains.

Acknowledgments

We thank our colleagues and students in the Design and use of Collaborative Telelearning Artifacts (DoCTA 1&2) projects (www.intermedia.uib.no/docta) at the University of Bergen and University of Oslo for contributing to the ideas and systems presented here. The projects were led by Barbara Wasson (1&2) and Sten Ludvigsen (2). We particularly wish to acknowledge the contributions of Silje Jondahl (CoPAS experiment), Karianne Omdahl (DoCTA pilot study) and Weiqin Chen (SA-Agent). Financial support was provided by ITU (National Network for IT-Research and Competence in Education), UFD (Norwegian Ministry of Education and Research), and InterMedia, University of Oslo. The third author was a graduate student at the Department of Informatics, University of Oslo while part of this work was performed. This paper was written while the first author was a visiting associate professor at CITE (Centre for Information Technology in Education), The University of Hong Kong.

References

1. Arnseth, H-C., Ludvigsen, S., Wasson, B. and Mørch, A. Collaboration and Problem Solving in Distributed Collaborative Learning. *Proceedings of EuroCSCL*, Maastricht, the Netherlands, 2001, pp. 75-82.

2. Bannon L. Computer-Mediated Communication. In: Norman DA, Draper SW, ed. *User Centered System Design: New Perspectives on Human-Computer Interaction.* Hillsdale, NJ, USA: Lawrence Erlbaum, 1986:433-456.

3. Beaudouin-Lafon M, Karsenty A. Transparency and Awareness in a Real-Time Groupware System. In: Green M, Mackinlay J, ed. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'92)*, Nov. 15-18, Monterey, CA, USA: ACM press, 1992:171-180.

4. Bratteteig T. Mutual Learning: Enabling Cooperation in Systems Design. In: Braa K, Monteiro E, ed. *Proceedings of the 20th Information Systems Research Seminar in Scandinavia (IRIS'20)*, Aug. 9-12, Hankø, Norway: Department of Informatics, University of Oslo, 1997:1-20.

5. Chen W, Wasson B. An Instructional Assistant Agent for Distributed Collaborative Learning. In: Cerri, S, Gouarderes G, Paraguacu, F, ed. *Proceedings of Intelligent Tutoring Systems Conference (ITS 2002)*, June 2-7, Biarritz, France and San Sebastian, Spain: Springer Lecture Notes in Computer Science 2363, 2002: 609-618.

6. Craig SD, Gholson B, Driscoll D. Animated Pedagogical Agents in Multimedia Educational Environments: Effects of Agent Properties, Picture Features, and Redundancy. *Journal of Educational Psychology* 2002;94:428-434.

7. Constantino-González, M. & Suthers, D. (2001). Coaching Collaboration by Comparing Solutions and Tracking Participation, In P. Dillenbourg, A. Eurelings & K. Hakkarainen (eds.) *Proceedings EuroCSCL 2001,* Maastricht, The Netherlands: Maastricht McLuhan Institute, 2001, pp. 173-180.

8. Dahlbäck N, Jönsson A, Ahrenberg L. Wizard of Oz Studies - Why and How. In: Gray WD, Hefley WE, Murray D, ed. *Proceedings of the 1993 International Workshop on Intelligent User Interfaces (IUI 1993),* Jan. 4-7, 1993, Orlando, FL, USA: ACM Press, 1993:193-200.

9. DeSanctis G, Gallupe RB. A Foundation for the Study of Group Decision Support Systems. *Management Science* 1987;33(5):589-609.

10. Dillenbourg P, Baker M, Blaye A, O'Malley C. The Evolution of Research on Collaborative Learning. In: Spada E, Reiman P, ed. *Learning in Humans and Machine: Towards an Interdisciplinary Learning Science*. Oxford, England, UK: Elsevier, 1996:189-211.

11. Dolonen J, Chen W, Mørch A. Integrating Software Agents with FLE3. In: Wasson B, Ludvigsen S, Hoppe U, ed. *Proceedings of the International Conference on Computer Support for Collaborative Learning 2003* (*CSCL 2003)*, June 14-18, Bergen, Norway: Kluwer Academic, 2003:157-161.

12. Dourish P, Bellotti, V. Awareness and Coordination in Shared Workspaces. In: Greenberg S, ed. *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92)*, Oct. 31- Nov. 4, Toronto, Canada: ACM Press, 1992:107-114.

13. Ellis CA, Gibbs SJ, Rein GL. Groupware: Some Issues and Experiences, *Communications of the ACM* 1991;34(1):39-58Ellis, Gibbs & Rein, 1991.

14. Fischer, G., Girgensohn, A.: End-user Modifiability in Design Environments. *Proceedings Human Factors in Computing Systems (CHI'90).* ACM Press, New York (1990) 183–189.

15. Fischer, G., McCall, R., Mørch, A.: Design Environments for Constructive and Argumentative Design. *Proceedings Human Factors in Computing Systems (CHI'89).* ACM Press, New York (1989) 269-275.

16. Fle3: Future Learning Environment. Website hosted by Media Lab, University of Art and Design Helsinki. Retrieved (2004-07-07) at http://fle3.uiah.fi/

17. Fronter: ClassFronter Virtual Learning Environment. Website hosted by Fronter Company. Retrieved (2004-07-07) http://fronter.info/uk/

18. Fugelli, P. Grounding in a CSCL Environment. Masters thesis. Faculty of Education, University of Oslo, 2004.

19. Grudin J. Computer-Supported Cooperative Work: History and Focus. *IEEE Computer* 1994;27(5):19-26.

20. Gutwin, C., Stark, G., Greenberg, S. (1995). Support for Workspace Awareness in Educational Groupware. In: *Proceedings of the ACM Conference on Computer Supported Collaborative Learning (CSCL'95*), Oct. 17-20, Bloomington, Indiana, USA: ACM Press, 1995:147-156.

21. Hakkarainen K, Lipponen L, Järvelä S. Epistemology of Inquiry and Computer-Supported Collaborative Learning. In: Koschmann T, Hall H, Miyake N, ed. *Computer Supported Collaborative Learning 2: Carrying Forward the Conversation*. Mahwah, NJ, USA: Lawrence Erlbaum, 2002:129-156.

22. Hara, N., Kling, R. Students' Frustrations with a Web-based Distance Education Course. *First Monday, 4* (12), 1999. Peer reviewed journal on the Internet. URL: http://firstmonday.org/issues/issue4_12/hara/index.html

23. Hiltz SR, Turoff M. Structuring Computer-Mediated Communication Systems to Avoid Information Overload. *Communications of the ACM* 1985;28(7):680-689.

24. Hughes JA, Randall D, Shapiro D. Faltering from ethnography to design. In: Greenberg S, ed. *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92)*, Oct. 31- Nov. 4, Toronto, Canada: ACM Press, 1992:115-122.

25. Johnson, W.L. and Rickel, J. Steve: An Animated Pedagogical Agent for Procedural Training in Virtual Environments. SIGART Bulletin 8 (1998) 16-21

26. Johnson WL, Rickel JW, Lester JC. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive learning Environments. *International Journal of AI in Education* 2000;11:47-78.

27. Johnson-Lenz P, Johnson-Lenz T. Groupware: The Process and Impacts of Design Choices. In: Kerr EB, Hiltz SR, ed. *Computer Mediated Communication Systems: Status and Evaluation*. New York, USA: Academic Press, 1982:44-55.

28. Jondahl S. and Mørch A. Simulating Pedagogical Agents in a Virtual Learning Environment. *Proceedings of the 24th Information Systems Research Seminar in Scandinavia (IRIS 24)*. Bergen: Department of Information Science, University of Bergen, 2001, 15-28.

29. Lazonder A.W., Wilhelm P. and Ootes S.A.W. Using sentence openers to foster student interaction in computer-mediated learning environments. *Computers and Education*, vol. 41, no. 3, 2003, 271-290.

30. Law, N. Scaffolding Scientific Conceptualization: Multiple Representation and Multilevel Visualization Using an Iconic Modeling Tool. *Proceedings ICCE 2002*, Auckland, New Zealand, IEEE Press, pp. 257-261.

31. Koschmann T. Paradigm Shifts and Instructional Technology: An Introduction. In: Koschmann T, ed. *Computer Supported Collaborative Learning: Theory and Practice of an Emerging Paradigm*. Mahwah, NJ, USA: Lawrence Erlbaum, 1996:1-23.

32. Krogsæter, M., Oppermann, R., and Thomas, C.G.: A User Interface Integrating Adaptability and Adaptivity. In: Oppermann, R. (ed.): Adaptive User Support. Lawrence Erlbaum, Hillsdale, NJ, 1994.

33. Lieberman, H.: Integrating User Interface Agents with Conventional Applications. *Proceedings Intelligent User Interfaces 1998*, pp. 39-46.

34. Ludvigsen S, Mørch A. Categorisation in Knowledge Building: Task-specific Argumentation in a Co-located CSCL Environment. In: Wasson B, Ludvigsen S, Hoppe U, ed. *Proceedings of the International Conference on Computer Support for Collaborative Learning 2003* (*CSCL 2003)*, June 14-18, Bergen, Norway: Kluwer Academic, 2003:67-76.

35. Maes P. Agents that Reduce Work and Information Overload. *Communications of the ACM* 1994;37(7):31-40.

36. Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., Cohen, M.D.: Intelligent Information Sharing Systems. *Communications of the ACM* 30, 5, 1987, 390-402.

37. Malone, T.W., Lai, K-Y., Fry, C.: Experiments with Oval: A Radically Tailorable Tool for Cooperative Work. *ACM Transactions on Information Systems* 13, 2, 1995, 177-205.

38. McManus, M.M., and Aiken R.M. Monitoring Computer Based Collaborative Problem Solving, *Journal of Artificial Intelligence in Education,* 6(4), 1995, 308-336.

39. Muukkonen H, Hakkarainen K, Lakkala M. Collaborative Technology for Facilitating Progressive Inquiry: Future Learning Environment Tools. In: Hoadley C, Roschelle J, (ed.). *Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 conference*. Dec. 12-15, Palo Alto, CA, USA: Lawrence Erlbaum, 1999:406-415.

40. Mørch, A. Three Levels of End-User Tailoring: Customization, Integration, and Extension. In: Kyng, M., Mathiassen, L. (ed.). Computers and Design in Context. MIT press, Cambridge, MA, 1997, 51–76.

41. Mørch A, Dolonen J, Omdahl K. Integrating Agents with an Open Source Learning Environment. *Proceedings of International Conference on Computers in Education 2003 (ICCE 2003)* (Hong Kong, China, Dec. 2-5, 2003). AACE Press, Norfolk, VA, 2003, 393-401.

42. Mørch, A. and Nævdal, J.E. Helping Users Customize Their Pedagogical Agents. *Proceedings Eighth Int'l Conf. on Knowledge-Based Intelligent Information & Eng. Systems (KES 2004)*, 2004, in press.

43. Nardi, B. A Small Matter of Programming. MIT Press, Cambridge, MA, 1993.

44. Ogata, H., Matsuura, K. and Yano, Y. Visualizing Knowledge Awareness in a Web-Based CSCL Environment. *Proceedings WebNet 2001*, pp. 927-932.

45. Roseman M, Greenberg S. TeamRooms: Network Places for Collaboration. In: *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work (CSCW'96)*, Nov. 16-20, Boston, MA, USA: ACM Press, 1996:325-333.

46. Rosenberg, M.J. *E-Learning: Strategies for Delivering Knowledge in the Digital Age*. McGraw-Hill, New York, NY, 2001.

47. Salomon G. What Does the Design of Effective CSCL Require and How Do We Study its Effects? *SIGCUE Outlook* 1992;21(3):62-68.

48. Scardamalia M, Bereiter C. Computer Support for Knowledge-Building Communities. *The Journal of the Learning Sciences* 1994:3(3):265-283.

49. Soller A, Linton F, Goodman B, Lesgold A. Toward Intelligent Analysis and Support of Collaborative Learning Interaction. *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, June, LeMans, France: IOS Press, 1999:75-82.

50. Stahl, G. A Model of Collaborative Knowledge-Building. In B. Fishman & S. O'Connor-Divelbiss (Eds.), *Proceedings of the Fourth International Conference of the Learning Sciences*, Mahwah, NJ: Erlbaum, 2000, pp. 70-77.

51. Suchman, L. *Plans and Situated Actions*: *The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, UK, 1987.

52. Svensson, L. (2002). *Communities of Distance Education*. Doctorial dissertation. Department of Informatics. Göteborg University, Sweden.

53. Terveen, L.G., Murray, L.T.: Helping Users Program Their Personal Agents. Proceedings Conference on Human Factors in Computing Systems (CHI' 96). ACM Press, New York, 355-361

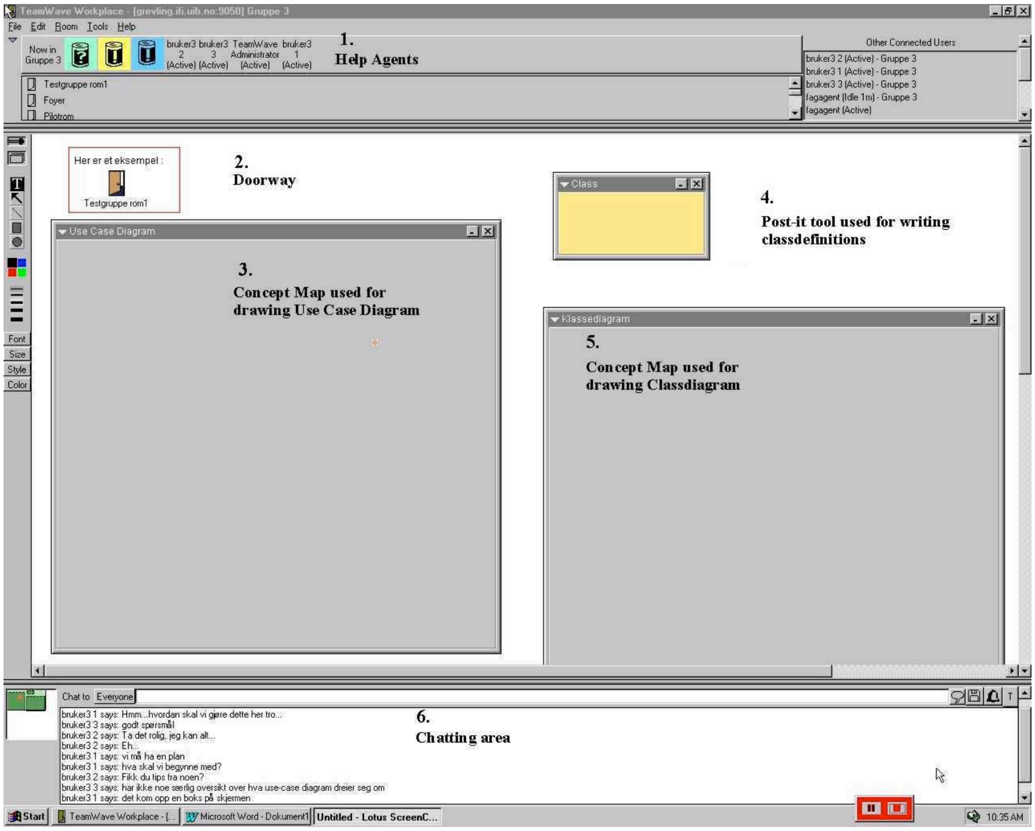
54. Toth, J., Suthers, D. and Weiner, A. Providing Expert Advice in the Domain of Collaborative Scientific Inquiry. *Proceedings Artificial Intelligence in Education (AIED'97),* August 20-22, 1997, Kobe, Japan, pp. 302-308.

55. Vygotsky, L. S. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press.

56. Wasson, B., and Ludvigsen, S. *Designing for Knowledge Building*. ITU report series vol. 19. Unipub (University of Oslo Press), Oslo, 2003.

57. Wasson B. and Mørch A. Identifying Collaboration Patterns in Collaborative Telelearning Scenarios. *Educational Technology & Society* 2000;3(3). Peer-reviewed journal on the Internet. URL: http://ifets.ieee.org/periodical/vol_3_2000/c04.html

58. WebCT: Introducing WebCT Vista 3.0. Website hosted by WebCT, Inc. Retrieved (2004-07-10) at http://www.webct.com/



TeamWave Workplace - [grevling.ifi.uib.no:9050] Gruppe 3
File Edit Room Tools Help
Now in Gruppe 3
bruker3 2 (Active)
bruker3 3 (Active)
TeamWave Administrator (Active)
bruker3 1 (Active)
Testgruppe rom1
Foyer
Pilotrom
Other Connected Users
bruker3 2 (Active) - Gruppe 3
bruker3 1 (Active) - Gruppe 3
bruker3 3 (Active) - Gruppe 3
fagagent (Idle 1m) - Gruppe 3
fagagent (Active)
Her er et eksempel :
Testgruppe rom1
Class
Use Case Diagram
Klassediagram
Font
Size
Style
Color
Chat to
Everyone
Start
TeamWave Workplace - [...
Microsoft Word - Dokument1
Untitled - Lotus ScreenC...
10:35 AM

1.
Help Agents

2.
Doorway

3.
Concept Map used for drawing Use Case Diagram

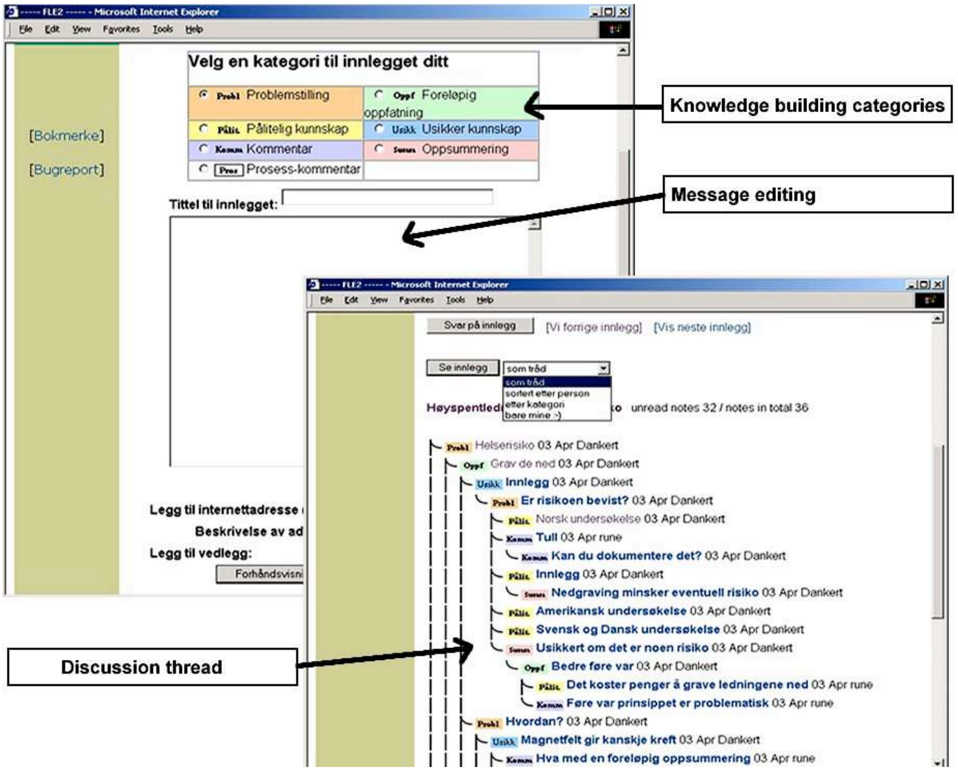
4.
Post-it tool used for writing classdefinitions

5.
Concept Map used for drawing Classdiagram

6.
Chatting area

bruker3 1 says: Hmm...hvordan skal vi gjøre dette her tro...
bruker3 3 says: godt spørsmål
bruker3 2 says: Ta det rolig, jeg kan alt...
bruker3 2 says: Eh...
bruker3 1 says: vi må ha en plan
bruker3 1 says: hva skal vi begynne med?
bruker3 2 says: Fikk du tips fra noen?
bruker3 3 says: har ikke noe særlig oversikt over hva use-case diagram dreier seg om
bruker3 1 says: det kom opp en boks på skjermen

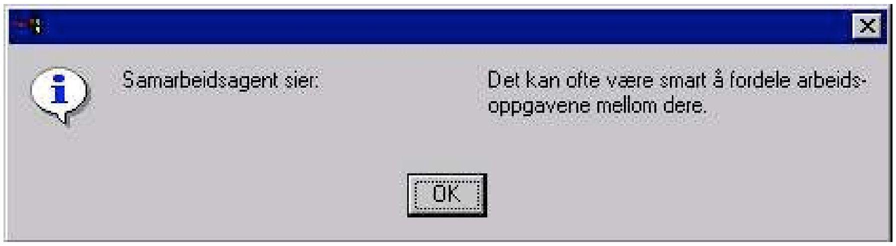
Figure 1: TW tools used in the CoPAS experiment. The Concept maps were used for creating design diagrams and the Post-it notes for class definitions. The chatting area was used for communication and coordination. Three pedagogical agents are represented as icons in the top left corner.



Høyspentled ko unread notes 32 / notes in total 36

- Probl Helserisiko 03 Apr Dankert
  - Oppf Grav de ned 03 Apr Dankert
    - Usikk Innlegg 03 Apr Dankert
      - Probl Er risikoen bevist? 03 Apr Dankert
        - Pålit Norsk undersøkelse 03 Apr Dankert
        - Komm Tull 03 Apr rune
          - Komm Kan du dokumentere det? 03 Apr Dankert
        - Pålit Innlegg 03 Apr Dankert
          - Summ Nedgraving minsker eventuell risiko 03 Apr Dankert
        - Pålit Amerikansk undersøkelse 03 Apr Dankert
        - Pålit Svensk og Dansk undersøkelse 03 Apr Dankert
        - Summ Usikkert om det er noen risiko 03 Apr Dankert
          - Oppf Bedre føre var 03 Apr Dankert
            - Pålit Det koster penger å grave ledningene ned 03 Apr rune
            - Komm Føre var prinsippet er problematisk 03 Apr rune
    - Probl Hvordan? 03 Apr Dankert
      - Usikk Magnetfelt gir kanskje kreft 03 Apr Dankert
        - Komm Hva med en foreløpig oppsummering 03 Apr rune

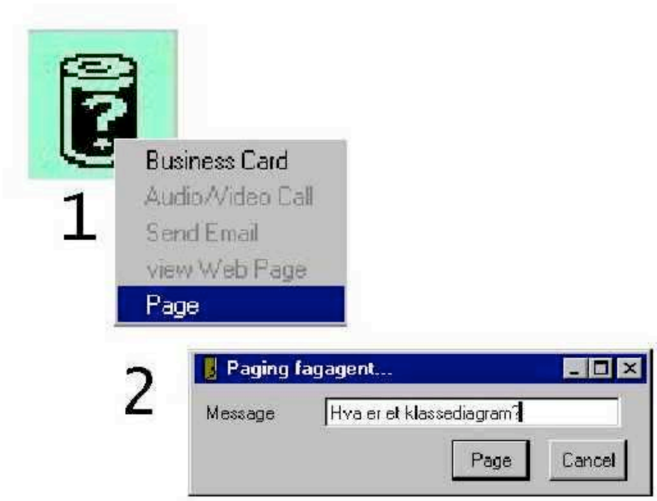
Figure 2: Future Learning Environment (Knowledge Building forum of Fle2): The leftmost window shows the writer's interface and the rightmost window the reader's interface.



Samarbeidsagent sier:

Det kan ofte være smart å fordele arbeidsoppgavene mellom dere.

Figure 3: A pop-up box from the Collaboration agent shows a message regarding division of labor. The message is written in Norwegian and reads: "It can be useful to divide the work." This piece of advice was presented when prolonged work by many users on the same shared object did not produce substantial change.



Message Hva er et klassediagram?

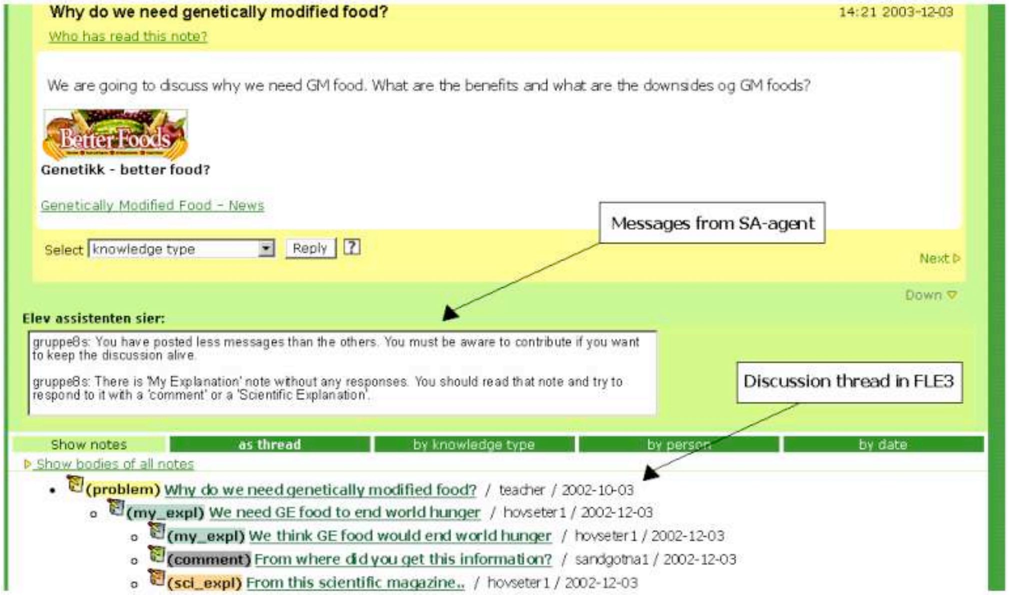
Figure 4: Steps for querying the Domain agent. The message in the TW "page" dialog reads: "What is a class diagram?"

CoPAS wizards	Message alternatives	Message actually used	Number of messages sent to all five groups
Tool Agent	23 *(12 new)*	82.5%	42
Domain Agent	19 *(6 new)*	63%	34
Collaboration Agent	15 *(1 new)*	86.5%	56
All wizards	57 *(19 new)*	77%	132

Table 1: CoPAS agent message types and number of times they fired. The *new* items refer to the number of messages created during the course of the experiment.

Fle2	Fle3	Intent
Problem	Problem	Identify problem or research question
My Working Theory	My Explanation	Personal hypothesis to address the problem
Deepening Knowledge	Scientific Explanation	Argue for hypothesis by finding pros and cons
Meta-comment	Evaluation of the process	Problematize the knowledge-building process
Comment	Comment	Comment on someone's previous posting
Summary	Summary	Summarise the knowledge-building activity

Table 2: Knowledge-building categories in FLE are based on the Progressive Inquiry model.



Why do we need genetically modified food? 14:21 2003-12-03

Who has read this note?

We are going to discuss why we need GM food. What are the benefits and what are the downsides og GM foods?

Genetikk - better food?

Genetically Modified Food - News

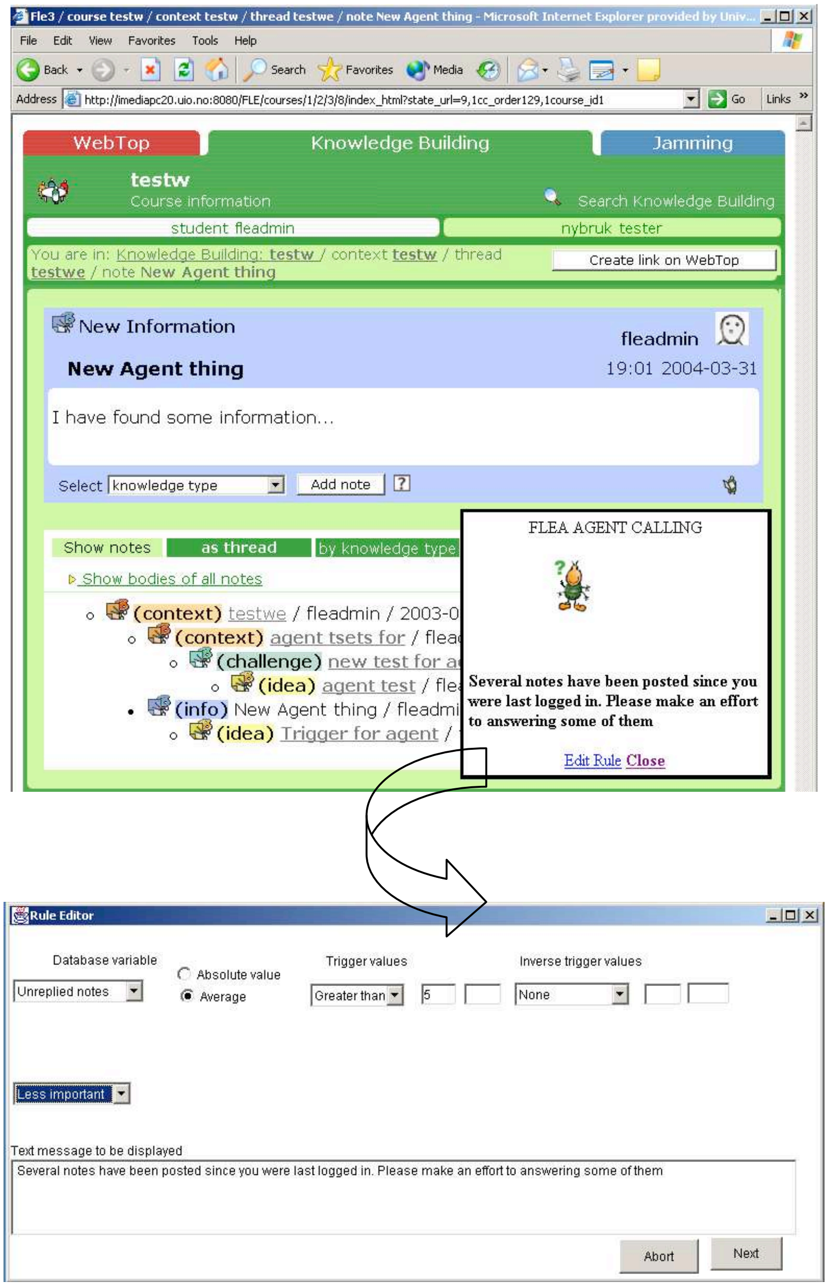
Elev assistenten sier:

gruppe8s: You have posted less messages than the others. You must be aware to contribute if you want to keep the discussion alive.

gruppe8s: There is 'My Explanation' note without any responses. You should read that note and try to respond to it with a 'comment' or a 'Scientific Explanation'.

- (problem) Why do we need genetically modified food? / teacher / 2002-10-03
  - (my_expl) We need GE food to end world hunger / hovseter1 / 2002-12-03
    - (my_expl) We think GE food would end world hunger / hovseter1 / 2002-12-03
    - (comment) From where did you get this information? / sandgotna1 / 2002-12-03
    - (sci_expl) From this scientific magazine.. / hovseter1 / 2002-12-03

Figure 5: FLE Student Assistant Agent. Messages from the agent are displayed in the window above the threaded discussion. The last message reads: "There is a 'My explanation' note without any response. You should read that note and try to respond to it with a 'Comment' or a 'Scientific Explanation'."



Fle3 / course testw / context testw / thread testwe / note New Agent thing - Microsoft Internet Explorer provided by Univ...

Address http://imediapc20.uio.no:8080/FLE/courses/1/2/3/8/index_html?state_url=9,1cc_order129,1course_id1

testw
Course information

student fleadmin

nybruk tester

You are in: Knowledge Building: testw / context testw / thread testwe / note New Agent thing

New Information

fleadmin

New Agent thing

19:01 2004-03-31

I have found some information...

Select knowledge type

- (context) testwe / fleadmin / 2003-0
  - (context) agent tsets for / flea
    - (challenge) new test for a
      - (idea) agent test / fle
- (info) New Agent thing / fleadmi
  - (idea) Trigger for agent /

FLEA AGENT CALLING

Several notes have been posted since you were last logged in. Please make an effort to answering some of them

Rule Editor

Database variable		Trigger values			Inverse trigger values		
Unreplied notes	○ Absolute value ● Average	Greater than	5		None		

Less important

Text message to be displayed

Several notes have been posted since you were last logged in. Please make an effort to answering some of them

Figure 6: FLE (top window) with SA-Agent superimposed as pop-up window (FLEA). Bottom window shows first screen of RuleEditor when called from the SA-Agent. The FLE discussion forum is translated into English for illustrative purposes; it does not reflect the Progressive inquiry model.