

An Agent System to Support Student Teams Working Online

Janice Whatley

Information Systems Institute, University of Salford, Manchester, UK

J.E.Whatley@salford.ac.uk

Abstract

In this paper an application of software agents is described, aimed at supporting students working on team projects in the online situation. Online teamwork is problematical for a number of reasons, such as getting acquainted with team members, communications between members and knowing what progress has been made on the project. Software agents have the ability to monitor progress and offer advice by operating in the background to act autonomously when the need arises. An agent prototype has been developed to perform a limited set of functions to support students, and the results of a trial carried out using teams working on projects are discussed.

Keywords: software agents, teamwork, online learning, task allocation, Prolog

Introduction

The higher education sector is being encouraged to provide more teaching materials and modules online, both as part of distance learning provision and as supplementary aids to learning for campus based courses (Eisenberg, 1998). Indeed technology has the potential to change the ways in which we teach and support students in the traditional university beyond recognition (Laurillard, 1993). There are difficulties in providing online tutorial support for students, and a particular problem is how students online can gain the same learning experiences as traditional campus based students (Thomas, Carswell et al., 1998).

The application of software agents to various online tasks has led to research into the ways in which agents may be used to support students online. In particular software agents may be used to support students undertaking group projects online. In this paper an agent system for supporting the maintenance tasks of group projects is described. The structure of a prototype system to support the planning stages of a group project is given, based upon research into problems students experience when carrying out group projects face to face.

Software Agents

The concept of an agent originates from human agents that provide services, such as estate agents and

travel agents. These agents have specialist skills, access to relevant information, contacts for obtaining information and are focused on a particular task. In the same way software agents are autonomous systems that work on behalf of a user (Bradshaw, 1997). They exhibit the ability to recognise what the user needs to accomplish and re-

Material published as part of these proceedings, either on-line or in print, is copyrighted by Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the publisher at Publisher@InformingScience.org

acts to the user's input. A more formal definition is:

An agent is a self-contained, concurrently executing software process, which encapsulates the current state in terms of knowledge, and is able to communicate with other agents through message passing (Wooldridge, 1995).

A software agent may operate in isolation, working on behalf of an individual, but their power derives from an ability to communicate with other agents to fulfil tasks they would be unable to complete alone. Typically a multi-agent system may consist of several agents, each capable of performing a different task autonomously. A network of agent systems, communicating over a wide area network (WAN) or a local area network (LAN), will make use of Internet connectivity to pass messages between each other. These multi-agent systems are the main thrust of current research, and have arisen as a result of the massive global infrastructure of networks now available.

Agent technology is a relatively new field of applying artificial intelligence (AI) to practical areas, e.g. knowledge management (Ferneley and Berney, 1999) and Internet searchbots (Lieberman, 1997). There are several examples of software agents acting as Internet search bots, such as Phibot or MySpiders, some combine the search facilities provided by several search engines into a more powerful search agent, attempting to reduce the information overload potentially experienced by people performing searches on the Internet (Henninger, 2002; Pant and Menczer, 2002).

Online Learning

The potential for using the Internet and the multimedia capabilities of technology for learning is great. Benefits may include provision for disadvantaged students as well as cost savings through economies of scale or automation of the teaching processes, also embracing video, audio and animation may help the learning process (Stephenson, 2001). E-learning is a term applied to systems for distance learning (Rudenstein, 1998), software to support students taking a campus-based course, or simply online documentation for teaching (Thomas, Carswell et al, 1998; O'Hagan, 1998). Online learners rely on Internet connections to communicate with institutions, tutors and other learners, and there is often a sense of isolation from the support of others .

There is increasingly too much information being made available for many of us to access and use appropriately (Hill & Raven, 2000). Where students are taking a campus-based course tutors are available, to guide students in organising their work and with suggested reading. However, online students often lack this tutor support (Mills & Tait, 1996). There is a need for help to find the best information for a given purpose, and to find it efficiently (Boud & Feletti, 1997).

In the field of e-learning software agents have the potential to help online learners in several ways. One such way is improving the effectiveness of searching (Ferneley & Berney, 1999), a system has been developed to share resources between students who have similar interests. Another aims to bring together students with similar interests or needs into a discussion area where they can receive help on particular problems (Vassileva & Deters, 2001). There are agents for guiding students in completing work, by offering tutorial help using a character (Nijholt, 2001). Finally, software agents may be used to help teach learners, for example using virtual environments to portray an example scenario (Aylett, 2001). Software agents can be made to work actively and adapt to users, which means they can simulate some of the roles of tutors. Pedagogical agents can monitor progress, give instruction when needed, help organise students' work and provide feedback for tutors .

Students Working on Team Projects

Traditional undergraduate campus-based courses incorporate a team project element, as an essential means of “learning by doing”. The learning cycle by Kolb (Kolb, 1984) summarises the stages of experiential learning as concrete experience, reflective observation, conceptualisation and active experimentation, which can be applied to student learning. This gives a starting point for thinking about how we approach the design of learning activities to achieve the learning outcomes. The main feature is that students do not learn by simply being told facts. They need to be able to practice using the facts, and reflect on the way they are used in order to form connections in the brain, which can be regarded as knowledge. Further experimentation, experience and reflection leads to intelligence or expertise in a subject. If the students are able to talk about this information, then they can be said to have knowledge of the subject, and intelligence shows in their ability to apply the knowledge in a variety of situations. Team projects give students an opportunity to discuss their understanding of the subject with their peers, as they apply the theory to practice (Sharan, 1990). Students undertaking online courses should be given a similar opportunity to experience team working, but where face to face contact is not possible, technologies may be able to provide additional resources to make the online team experience comparable.

Computer mediated communication (CMC) tools, such as conferencing, email and discussion forums support the communication needs for the task roles of team projects, examples of their use are given in (English and Yazdani, 1998) and (Hendson, 1997). The facilities included in Virtual Learning Environments (VLE) give students the capability to communicate with each other and the tutors, and are based to a large extent on the facilities incorporated in Groupware products, which in turn have been developed as a result of research into Computer Supported Cooperative Working (CSCW) (Connolly, 1994). The VLE’s provide a structure to enable communication, but little help in the process of communication to help the students form workable learning networks (Lawther and Walker, 2001). Opie used the term “knowledge-based teamwork” to describe the sort of interaction between team members who are all bringing to the case in hand their own interpretation of the situation, through their own knowledge or expertise. Her work is specifically related to health care, but this is a typical domain in which teamwork is essential for achieving outcomes (Opie, 2000).

At this point it is useful to distinguish between a “group” and a “team”. Belbin gave a useful description of both (Belbin, 2000), with groups comprising any number of members, and as their size increases individual contributions tend to be reduced, also there is often a hierarchical structure to a group. On the other hand a team is usually smaller, has shared objectives, with each member considering how best to contribute, and often imprinting their personal identity in the social setting of the team.

Successful team working requires that the maintenance roles as well as the task roles of the team are given attention (Hartley, 1997). Group dynamics play an important role in determining how successful the outcome of the project is, i.e. the ways in which the members interact with each other and how this changes with time as the team develops (Bion, 1961), (Gibbs, 1994), (Jaques, 1984). Gilly Salmon (Salmon, 2000) suggested ways in which tutors can help students to interact socially online, in order to develop team cohesion. Student support using commercial groupware products enables communication between team members and instructors (Tiwari and Holtham, 1998), also BSCW is an example of a tool that has been used as support for team projects and was found useful for information sharing, offering greater flexibility in students’ face to face communication, but it offers limited support for the maintenance roles of teamwork (Vliem, 1998). In previous work, students’ perceptions of the manner in which their team worked together confirmed that teams were more likely to be successful in their projects if they pay attention to some of the maintenance factors (Whatley, Beer et al, 1999).

In this paper an application of a software agent for supporting students working on team projects online is described. The support needed by students for teamwork differs from that which might be appropriate for an individual working alone, as the dynamics of team working also need to be considered. The ad-

vantage of using software agents for supporting online students is that agents can bridge the divide between time and place. Students may be dispersed and working at times to suit themselves, so the agents can keep track of the students' progress on the work, and enable all students to be aware of the status of the project.

An action research approach was adopted for this study, because a more user-centred design may be achieved by active user involvement in the development process. Over several iterations of a prototyping method, further functions may be added and refined, by considering feedback from students in the form of questionnaires, interviews and focus groups. Although each successive cycle will not involve the same individuals, a broadly similar class of students will participate in the design process, so that the final product will be acceptable to a wide range of students.

Functionality of a Software Agent for Team Working

To see how technology can be applied to team projects, it is necessary to analyse the stages of a team project, and to determine the particular problems encountered at the different stages. After gathering questionnaire and interview data on the problems associated with team working in the face to face situation, we were

Project stage	Factors identified as problematical
Planning	Introductions Setting ground rules Produce a project plan Allocate tasks
Doing the project	Check the time schedule Ensure all members contribute Identify lack of skills Discuss each others' contributions
Completing	Collating the individual parts Preparing a report Appraising the team's performance

Table 1 –Stages of the team project.

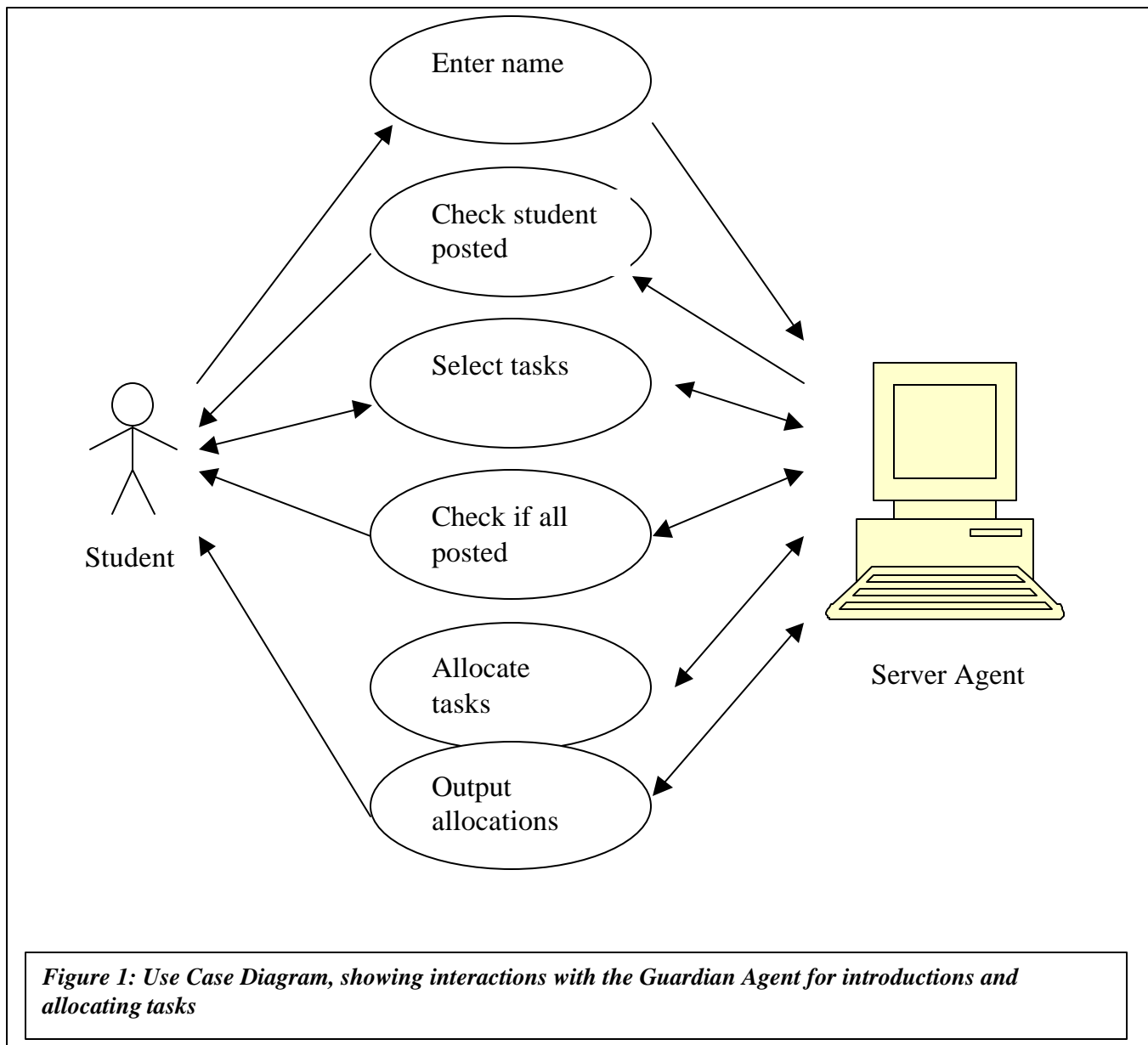
able to identify some of the factors that may contribute to the success or otherwise of team projects when transferred to an online situation (Whatley, Beer et al, 1999). A simplified summary of team project stages and some identified factors in given in Table 1.

These stages of a team project do not correlate directly with the stages of team development defined by Lawrence (Bligh, 1974), but represent stages of the tasks that students will identify with (O'Sullivan and Rice, 1996). The identified factors "introductions" and "setting the ground rules" are significant processes towards the maintenance roles of team projects. It was decided that the initial work on developing a software agent to support students, called a Guardian Agent, should be targeted at the functions associated with the planning stage of a project.

Design of the Prototype Agent System

The initial prototype for the Guardian Agent is being developed in LPA Prolog, using their Agent Development Kit (Logic Programming Associates, 2000). This tool enables the developer to code the interfacing aspects of the agent without worrying about the technicalities of the agent communication, which is dealt with by the tool. The declarative features of Prolog were used for handling facts and rules, which can be passed between each student's agent and the server agent.

In the chosen system structure, each individual student communicates with the agent system by means of his or her individual Guardian Agent. Each agent will have a similar structure when the project begins, with interfacing capabilities for communicating with its student, reasoning capabilities for monitoring and analysing the current situation, a knowledge base personal to its student and communication capabilities for communicating with other students' agents. All communications between agents is through a



server agent, allowing for a knowledge base to be built up for the particular project the students are working on.

The process of allocating roles begins with finding out about each other's abilities and preferences. Students working face to face introduce themselves to other members of their team and they tell them verbally what their abilities and preferences are. There may be several students working on the team project online, each accessing the project site at different times and not knowing which of the other students have already introduced themselves. The first function of the Guardian Agent is to determine whether or not its student has already posted their abilities and preferences. Where the agent finds that its student has not posted their abilities and preferences, the agent asks its student to identify the predetermined task areas he or she likes, is good at, dislikes and is not good at.

The Guardian Agent can obtain its own student's abilities and preferences and post these to the server agent so that all of the students' agents can access them. Once all of the students in the team have posted their abilities and preferences the agent system can apply a set of rules to the facts, in order to determine which tasks of the project could be allocated to each student. The agent system will maintain a record of the suggested allocations on the server agent. As each student returns to the project task, the agent will

An Agent System to Support Student Teams

present the allocations, so that the student can consider and discuss them with the other students on the project. Figure 1 shows the use case diagram for the interactions with the agent system for the introductions and allocation of tasks functions.

A typical interface window used to obtain a student's abilities and preferences is given in Figure 2. The tasks are predetermined by the tutor for the particular project, and stored by the server agent, for access as required by the individual agents.

The Guardian Agent has been programmed to work with three levels of allocation, using the following rules:

Allocation1 -

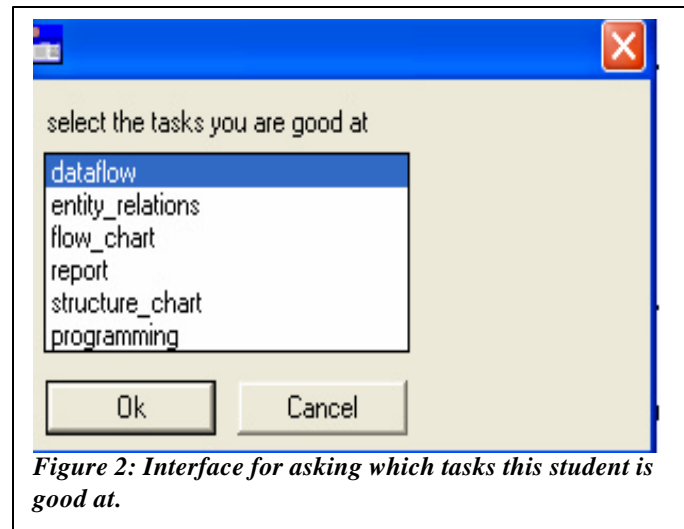
If studentA likes X and is able at X
Then studentA should do X.

Allocation of tutoring -

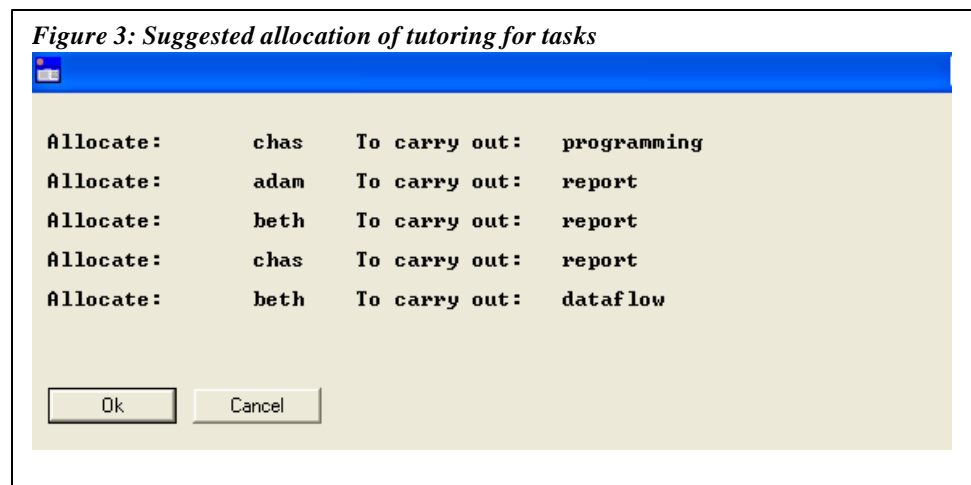
If studentB likes X, but is unable at X
Then studentB could be offered tutoring in X

Allocation2 -

If studentC is good at X and has not expressed a dislike of X
Then studentC could do X



An example of the text window indicating suggested task allocations is shown in Figure 3. The rules are built into the Guardian Agent, but an additional enhancement will be that the tutor can choose which ruleset to apply to a particular team project. In this way if the tutor requires that students should gain experience in a task they have not



previously undertaken, this can be changed by selecting an appropriate ruleset.

Students interact with their Guardian Agent according to the cases shown in Table 2, depending upon the stage of their use of the Guardian Agent and the stage of the project. For example in Case 2 the student has not posted their abilities and preferences, so the Guardian Agent obtains these from its student, and posts them to the server agent. Next the Guardian Agent consults the server agent to determine whether all the other members of the team have already posted theirs. When the agent finds that all of the abilities and preferences have been posted, the program to suggest allocations of tasks and tutoring is run, and the results are shown to the student and posted to the server agent. As each student may be accessing the project site at different times, the status of the project can be returned as required by the student, to keep the student up to date.

Table 2: Cases for student interaction with the agent system

	Student not posted	Student posted	Not all students posted	All students posted	Action
Case 1	*		*		Get preferences, wait for other students
Case 2	*			*	Get preferences, run the allocations
Case 3		*	*		Wait for other students
Case 4		*		*	Allocations retrieved

A shell program was then written to allow for all cases of interaction between the student and the Guardian Agent. This program works by consulting the server agent to determine the status of the project, then continuing according to the stored facts.

Students working on the team project will have at their disposal a variety of CMC tools, including email and discussion forums. It is likely that there will be some adjustments needed to the suggested allocations, for instance if too many students have been allocated to the same task (as on Figure 3), or no students have been allocated or offered training for a particular task. After informing the students of the need for adjustment, the students would be encouraged to discuss the allocations given, and agree to accept them or negotiate alternative allocations. Indeed the allocations to tasks are only a suggestion, the students in the team are free to accept or reject the advice of the agents. The agent system has simply removed a time consuming round of questioning and analysing, which is a fairly straightforward process in the face to face setting, but more cumbersome online.

The allocation of tutoring is a means to determine whether a student requires access to some additional tutoring material. The project site may be populated with a selection of tutoring material, which may be suggested at the planning stage of the project, or may be suggested later on as the project progresses, when the agent recognises that its student has not produced a planned piece of work, due to a lack of the appropriate skills.

Next the students should agree a set of ground rules they will all work to, the Guardian Agent will prompt for the student to input suggestions, which will be stored on the server agent. These will be posted to the server agent as text strings, from where they can be retrieved by one of the Guardian agents and collated as a discussion list. Similarly a project plan should be devised, once again the Guardian agent prompts its student to suggest tasks that need to be performed and the time they will take. The agent system is now prompting for action and encouraging the students to formulate a project plan, but also enabling all of the students to be involved in the process.

The work on designing the Guardian Agent functionalities and interfaces has been completed. The limited function Guardian Agent prototype has been evaluated to determine to what extent it can enhance the effectiveness of online team project work in the ways intended, and we evaluated its acceptability to both staff and students. The second is most important as if the agent is not fully accepted and trusted by all parties it will not ultimately succeed in its goals. The results of the evaluation will be reported in the next section.

Results from Trial of Agent System

The Guardian Agent was tested with seven teams working on projects in systems development, as part of their undergraduate programme. The teams consisted of between 6 and 10 second and final year members, working on campus, and they were asked to use the allocation of tasks function as they began their projects. Each team project is slightly different, so the tasks were specific to each team. After some brief instructions for using the agent system, each student in the teams used the Guardian Agent to input their details over a period of four weeks. As not all students were present for each session, they did not

Table 3 – Results of the questionnaires completed by students

Questions to students after completing the Guardian Agent trial	Number answering “Yes”	% of total responses
Did you find the function useful?	20	56
Did you find the system easy to use?	32	89
Was it self explanatory?	28	78
Do you think it would be useful for students online?	29	81
Do you think it would be useful for students on campus?	23	64
Do you like the concept of agent help for working online?	27	75
Do you like the concept of agent help for working on campus?	22	61
Would you personally like to use this sort of agent?	20	56

all use the system on the same occasion, which matched the way in which the agent might be used online.

Afterwards the students were asked to complete questionnaires and were invited to a focus group so that we could obtain feedback on the usefulness of the system. A summary of the results from the questions asked is given in Table 3.

The interface was generally acceptable, but some students suggested improvements, which we can incorporate into future developments. About half of the students said that the output from the allocation of tasks function was useful, these were mainly team leaders, who compared the output with the ways in which they would have normally performed made selections. A majority of the students thought that such an agent system would be useful to students working online as well as for campus based students. Just over half of the students said that they personally would like to use such an agent. As part of the evaluation issues concerning the scalability of the system, integration of the system into a user interface and portability of the system to other platforms were considered. The prototype was used by seven teams, about 55 students in all, however, they did not all try to access the server agent at the same time. The prototype was running in the programming environment, so the interface was slower to operate than it should have been, leading to some dissatisfaction for the students. The speed of message passing between the Guardian Agents and the server agent was acceptable, using the internal network.

Conclusions

Team project work is important in many areas of higher education, especially in information systems development. In an age of global working students should have the opportunity to experience working online as part of their studies, agent support for online working is one possible means of helping students to learn from their experiences. The prototype Guardian Agent has been designed and tested within a Systems Analysis and Design module, but the system is intended to be applicable to many subject areas.

In this paper the design of a Guardian Agent system to support students working on team projects online has been described. The prototype system has been produced for the planning stages of a team project so far, and will be extended to cover other stages of a team project. Evaluation of the prototype has shown that the concept of an agent for supporting learning tasks is largely acceptable for students and tutors. There is considerable potential for agent systems to help students in their learning, particularly agents to help with team working when the students are widely dispersed.

In future cycles of this study improvements to the user interface will be considered, the system will be tested with a larger number of teams, and with teams working over the Internet. Additional functions will be added and evaluated for their acceptability to the student groups.

References

- Aylett, R. (2001). Intelligent virtual agents. *IVA2001*, London, Springer Verlag.
- Belbin, R. M. (2000). *Beyond the Team*. Oxford, Butterworth-Heinemann.
- Bion, W. (1961). *Experiences in groups*. London, Tavistock.
- Bligh, D. (1974). *What's the use of lectures*. London, Penguin.
- Boud, D. and G. Feletti (1997). *The challenge of problem-based learning*. London, Kogan Page.
- Bradshaw, J. (1997). *Software agents*. London, MIT Press.
- Connolly, J. (1994). *CSCW and artificial intelligence*. London, Springer Verlag.
- Eisenberg, D. (1998). College faculty and distance education. *Virtual University Journal*, Online VU. Issue 2.
- English, S. & M. Yazdani (1998). *Computer supported cooperative learning in a virtual university*. Internet, Virtual University.
- Ferneley, E. H. and B. Berney (1999). CASMIR: A Community of Software Agents Collaborating in Order to Retrieve to Multimedia Data. *3rd International ACM Conference on Autonomous Agents (Agents '99)*, Seattle, WA, USA.
- Gibbs, G. (1994). *Improving student learning through assessment and evaluation*. Oxford, Oxford Staff Dev.
- Hartley, P. (1997). *Group communication*. London, Routledge.
- Hendson, B. (1997). Groupwork with multi-media in maths: the role of the technology and teacher. *Br Journal of Educational Technology* 28, 4: 257 - 270.
- Henninger (2002). *Phibot*. <http://phibot.org/new/Search/About>
- Hill, J. R. and A. Raven (2000). Online Learning Communities: If You Build Them, Will They Stay? *ITForum*. 2002.
- Jaques, D. (1984). *Learning in Groups*. London, Kogan Page.
- Kolb, D. (1984). *Experiential learning*. London, Prentice Hall.
- Laurillard, D. (1993). *Rethinking university teaching*. London, Routledge.
- Lawther, P. and D. Walker (2001). An Evaluation of a Distributed learning System. *Education and Training* 43(2): 105-116.
- Lieberman, H. (1997). Autonomous Interface Agents. *Proceedings of the ACM Conference on Computers and Human Interface, CHI-97*, Atlanta, Georgia.
- Logic Programming Associates (2000). LPA. London. <http://www.lpa.co.uk/index.htm>
- Mills, R. and A. Tait (1996). *Supporting the learner in open and distance learning*. London, Pitman.
- Nijholt, A. (2001). Agent assistance: from problem solving to music teaching. *First International Workshop on Agents and Internet Learning*, Montreal, International Conference on Autonomous Agents.
- O'Hagan, C. (1998). The virtual university and the Cheshire Cat. *Universities in a digital era Vol 1*: 244 - 247.
- Opie, A. (2000). *Thinking teams/ thinking clients: knowledge-based teamwork*. New York, Columbia University Press.
- O'Sullivan, T. and J. e. a. Rice (1996). *Successful group work*. London, Kogan Page.
- Pant, G. and Menczer, F. (2002). MySpiders: Evolve your own intelligent Web crawlers. *Autonomous Agents and Multi-Agent Systems* 5(2): 221--229.
- Rudenstein, N. (1998). *The Internet and education: a close fit*.
- Salmon, G. (2000). *E-moderating*. London, Kogan Page.
- Sharan, S. (1990). *Collaborative learning: theory and research*. New York, Praeger Publishers.
- Stephenson, J. (2001). *Teaching and Learning Online: pedagogies for new technologies*. London, Kogan Page.

An Agent System to Support Student Teams

- Thomas, P., L. Carswell, et al. (1998). A holistic approach to supporting distance learning using the internet: transformation, not translation. *Br Journal of Educational Technology* 29, 2: 149 - 161.
- Tiwari & C. Holtham (1998). Learning Groupware through using groupware - computer supported collaborative learning with face to face students. *ITiCSE*, ACM.
- Vassileva, J. & R. Deters (2001). Lessons from deploying I-Help. *First International Workshop on Agents and Internet learning*, Montreal, International Conference on Autonomous Agents.
- Vliem, M. (1998). *Using the Internet in university education: the application of BSCW within student projects*. Twente, University of Twente.
- Whatley, J., M. Beer, et al. (1999). Group Project Support Agents for helping students work online. *HCI International 1999*, Munich, Germany, Lawrence Erlbaum, London.
- Wooldridge, M. J., N (1995). *Towards a theory of cooperative problem solving*.