SOFTWARE AGENTS FOR SUPPORTING STUDENT TEAM PROJECT WORK

Janice E. Whatley
Information Systems Institute, University of Salford, Manchester, M5 4WT,

Keywords: Software Agents, Teamwork, E-Learning,

Abstract: In this paper an agent system is described, which has been designed to support students undertaking team projects as part of their studies on campus or online. Team projects form an important part of the learning process for campus based students, but are not easily incorporated into the learning activities for online students. The particular problems of working on projects in teams are explored, and an agent system was designed to support some of the maintenance tasks of team working. Agent technology is suggested because of the ease of communication between software agents and their autonomy in operation. The agent system has been tested on student teams working on campus, and the results indicate that this type of support agent may be helpful to students. The modified version of the agent system was successfully implemented, and the trial suggests that it may be scaled up to use over the Internet to support online student teams.

1 INTRODUCTION

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 (Hill and Raven, 2000). Campus based learners are beginning to rely more on the Internet to support their studies, such as to enable them to access material outside of lecture times, to work at more convenient times and wherever they choose and to supplement their face to face contact with other students.

Working in teams is particularly problematical, both for campus based students and for online students, but team working, and virtual team working, especially, is becoming an essential skill for our graduates. Groupware and virtual learning environments help team members to communicate and share files, but do not support the maintenance needs of team working, which are necessary for successful team operation.

Artificial intelligence has been used to develop tutoring systems for individualised learning, and agent technology is being harnessed for Internet based communicating systems. The analogy between multi-agent systems and student teams has pointed to the possibility of agent technology as a solution to some of the difficulties of working in teams, by combining the benefits of intelligent tutoring, advice from an agent and communication.

2 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;
- active experimentation.

These stages give 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

E. Whatley J. (2004).
SOFTWARE AGENTS FOR SUPPORTING STUDENT TEAM PROJECT WORK.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 190-196
DOI: 10.5220/0002633201900196

their ability to apply the knowledge in a variety of situations. Collaborative learning may range from a pair of students working together to a large class of students learning together, but as Dillenbourg says, ther is no agreed definition of collaborative learning (Dillenbourg 1999). 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 may be working at times collaboratively and at times cooperatively, coordinating their efforts to achieve a project outcome, and learning about themsolves and about the subject in the process (Dillenbourg, Baker et al. 1996). When campus based students work on team projects they experience difficulties in organising their work together, such as arranging meetings, deciding who should carry out which part of the work, and coping with non participation from members. If we are to provide students undertaking online courses with a similar opportunity to experience team working, we need to provide some means of helping them to organise their work together. Where face to face contact is not possible, technologies may be able to provide some additional resources to help make the online team experience comparible to the campus based team experience.

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).

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 (Basic Support for Cooperative Working) 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 et al., 1999a).

The essence of learning how to work in a team is an important aspect of team projects, because organisations make much use of team working, whether face to face, or, increasingly, in a virtual team. The experience provided in Higher Education is important, but concentrates on face to face teams, whereas there is an increasing need to offer the opportunity to learn to work virtually as well.

3 ONLINE TUTORING WITH AGENTS

The Internet is providing possibilities for learners to access their course materials in a variety of ways. Some may prefer to use traditional face to face means of learning, whereas others may prefer to learn from home or the workplace, taking advantage of online access. The new breed of "blended learners" expect to be able to choose when and where to access their lecarning, and require support to enable them to learn effectively.

Intelligent tutoring using artificial intelligence (AI) concepts has been associated with distance learning, providing interfaces that respond to an individual user's needs (Farr and Psotka, 1992), though very much aimed at individuals using programmed learning packages. Agent technology is a relatively new field of applying AI to practical areas, e.g. knowledge management (Ferneley and Berney, 1999) and Internet searchbots (Lieberman, 1997). Knowledge management aims to enable collaboration between individuals online, notably for problem solving (Corkill, 2003). Virtual communities in the workplace are becoming more common, but collaborating globally requires different skills from those used for face to face collaboration (Lipnack and Stamps, 2000).

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 reacts 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).

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 or enable the sharing of resources between students who have similar interests (Ferneley and Berney, 1999). Another aims to bring together students with similar interests or needs into a discussion area where they can receive help on particular problems (Vassileva and 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). Soller suggests an architecture for multi-agents to support online group learning, concentrating on knowledge sharing between students (Soller and Busetta). 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 (Baggetun et al., 2002). These agent systems continuously operate in the background on a student's workstation and act autonomously to suggest ways in which the learner might improve performance.

A software agent may operate in isolation, working on behalf of an individual, similar to personalised intelligent tutors, 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.

We now turn to the notion that multi-agents may be applied to supporting collaborating members of a team, and in particular teams of learners. In the next section an application of a software agent for supporting students working on team projects is described.

4 DESIGNING A SOFTWARE AGENT SYSTEM FOR TEAM WORKING

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 advantage 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 the students to be aware of the status of the project. Similarly, campus based students may benefit from such a software agent system, as these students demand the flexibility to work at different times and places.

A preliminary version of an agent system prototype has been developed, performing a limited set of functions to help students to get started on their teamwork, and the results of a trial carried out using teams working on projects on campus are discussed. These results have informed our further design, leading to a second version of the agent system, which has similarly been tried on student teams working on campus.

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 range

Table 1: Stages of the team project

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

of students have participated in the design process, and the final product should be acceptable to a generic type of students.

5 FUNCTIONS OF THE AGENT SYSTEM

There are limited examples of teamwork being used for learning activities with online learners, because there is a belief that face to face contact is essential for successful teamwork (Lewis, 2002). Previous surveys have been carried out to determine the nature and extent of the difficulties experienced by students working face to face, the intention being to design an agent system to alleviate these problems initially (Whatley et al., 1999b). Three main stages of a team project have been identified, each with its own associated problems, and these are summarised in Table 1.

These stages of a team project do not correlate directly with the stages of team development originally defined by Bruce Tuckman (Tuckman, 1965), but represent stages of the tasks with which students will identify (O'Sullivan et al., 1996). The identified factors "introductions" and "setting the ground rules" are processes that contribute towards the maintenance roles of team projects. It was decided that the initial work on developing a software agent to support students should be targeted at these functions, forming part of the planning stage of a project.

6 DESIGN OF THE FIRST PROTOTYPE AGENT SYSTEM

The initial prototype for the agent system was developed in LPA Prolog, using their Agent Development Kit (Logic Programming Associates, 2000). This tool enabled 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. The first prototype considered the allocation of tasks to the team members.

In designing the prototype agent system, we were interested in these main features:

- whether such an agent system would be acceptable to students;
- whether the agent system would be of any help to the students;
- whether communicating agents could be implemented successfully within a typical intranet environment.

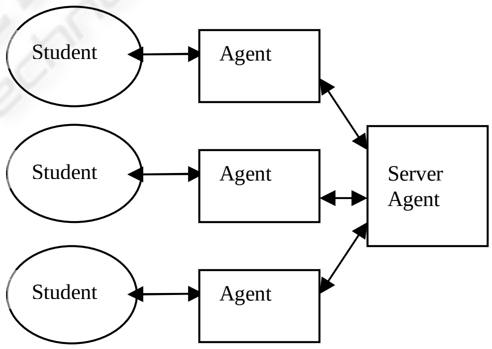

Figure 1: Structure of Guardian Agent System

In the chosen system structure, each individual student communicates with the agent system by means of their individual agent (Figure 1). Each agent will have a similar structure when the team 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.

Online students may be working on the team project at different times of the day, so there is limited possibility of discussing the allocation of the tasks for the project. Even campus based students may not all be present at the same time for formal meetings. Comparing individual abilities would take a considerable length of time using a discussion forum, or other CMC means. Thus the agent system aims to reduce the time spent on a mundane task, and inform the students of the status of this task.

The process of allocating tasks begins with the agent asking its student to enter details of their abilities and preferences. The agent system will 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, necessary later when the agent system will be able to monitor student activity against the work plan. As each student returns to the team project, the agent will present the allocations, so that all of the students can consider and discuss them with the other students on the project. Any allocations proposed by the agent system may be subject to negotiation between the students, the allocations are simply suggestions.

The agent system 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.

Allocation2 –

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

Allocation of tutoring -

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

7 RESULTS FROM THE FIRST TRIAL OF THE AGENT SYSTEM

The agent system 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 agent to input their details over a period of four weeks. As not all students were present for each session, they did not 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.

The interface was generally acceptable, but some students suggested improvements, which we have incorporated into the second prototype. 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 made task allocations. 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.

Table 2 – Questionnaire findings

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

From our initial results it is clear that students would find an agent system acceptable, particularly for online students (though the term "online" was not defined). It is difficult to establish how helpful the agent system might be to either online or campus based students, but comments made during the focus group session did suggest that the agent system performed a helpful function.

Most importantly, we did find that the agent system implementation, using a server agent, was successful, though a limited number of students were involved in the study. However, implementing the system using Prolog proved difficult, owing to the reluctance of the code to compile correctly.

8 DESIGN FOR THE SECOND TRIAL OF THE AGENT SYSTEM

Taking into consideration the feedback from students and issues of portability, we built the second prototype in Java. The programs for the agents were produced as Java executable files, incorporated into web pages, and an improved user interface was designed for the system. The server agent was replaced by a file server, running a MySQL database, in which the facts are stored. Figure 2 shows the screen to obtain a student's preferences.

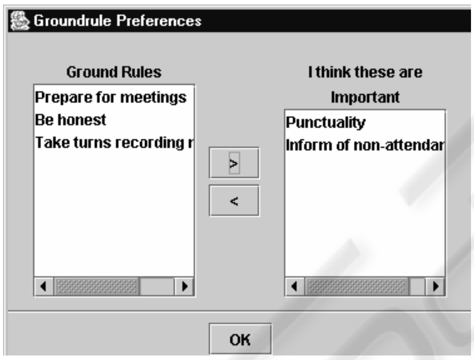

Figure 2: example of the interface to obtain preferences.

The issue of agreeing ground rules for team working has been little explored, and problems cited included difficulties getting students to attend meetings, inform the team leader if they cannot attend and complete their assigned work on time (Hill and Raven, 2000). So this additional functionality was included, to help the students to agree ground rules for the way they intend to work together. Figure 3 shows the interface for asking about ground rules.

The second prototype was tried on 25 teams of between 10 and 15 students, working on campus, and took place over five weeks, when theses teams were establishing their individual task areas for their projects.

Once again questionnaires and focus groups were used to capture the students' opinions, together with interviews carried out with the team leaders to

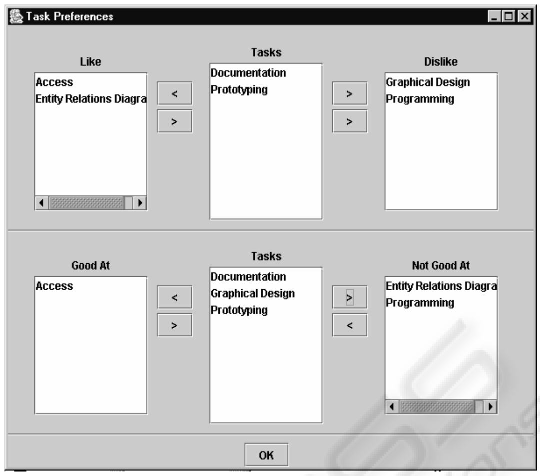

Figure 3: interface to ask about ground rules.

ascertain their views on the differences the agent system made to their team project.

9 RESULTS FROM THE SECOND TRIAL OF THE AGENT SYSTEM

Initial analysis of the findings from the second trial, indicates some satisfaction with the functions of the agent system, several team leaders said that the allocation of tasks was a useful function. Some limitations with the interface were identified and the pre-programmed task list did give rise to some reluctance to use the agent system, as one team leader said that the tasks included were not relevant to their particular project.

Technically, the agent system was a success, as the MySQL database was able to cope adequately with the number of students using the system. In spite of the fact that not all of the computers in the laboratory were equipped to run Java programs, sufficient machines were available to satisfy the demand for using the agent system at any one time. The pattern of usage for the campus based students probably matched the expected pattern for online teams.

Further development work will take place to improve the interface and add more functionality to the agent system.

10 CONCLUSIONS

In this paper we have described two phases in the development of an agent system for supporting students working on team projects. At present the system has only been tried with campus based students, but feedback from these students indicates that the results are likely to be applicable to online teams.

We have developed an agent system to help student teams to allocate the tasks between the members of the team, and to help the team members to agree to a set of ground rules. Although these are two relatively simple functions, and only a part of the planning stage of a project, our findings indicate that an agent system that can support students through their team project would be acceptable.

Future investigations will try to establish the extent to which student teams may be helped by an agent system to perform more successfully.

The mode of implementation of the second prototype agent system perhaps deviates from the true definition of an agent, as each instance of the agent does not maintain its own knowledge base. However, as the functionality of the agent system is enlarged in the future, it is suggested that a self-contained knowledge base might be an essential part of each agent for monitoring its student against a project plan.

Although this trial was carried out on campus, within a department's intranet, the results suggest that the Java agent can be distributed over the Internet. Issues of security and firewalls may need addressing, in preparation for it to be implemented for real online teams.

REFERENCES

Aylett, R. (2001) In *IVA2001*, Vol. LNAI 2190 Springer Verlag, London.

Baggetun, R., Dolonen, J. and Dragsnes, S. (2002).

Bion, W. (1961) *Experiences in groups*, Tavistock, London.

Bradshaw, J. (1997) *Software agents*, MIT Press, London.

Connolly, J. (1994) *CSCW and artificial intelligence*, Springer Verlag, London.

Corkill, D. (2003) In *International Lisp Conference*.

Dillenbourg, P. (1999). Collaborative learning: cognitive and computational approaches. Oxford, Elsevier.

Dillenbourg, P., M. Baker, et al. (1996). The evolution of research on collaborative learning. Learning in humans and machine: towards an interdisciplinary learning science. E. Spada and P. Reiman. Oxford, Elsevier**:** 189-211.

English, S. and Yazdani, M. (1998) Virtual University, Internet.

Farr, M. and Psotka, J. (1992) *Intelligent instruction by computer: theory and practice*, Taylor & francis, New York.

Ferneley, E. H. and Berney, B. (1999) In *3rd International ACM Conference on Autonomous Agents (Agents '99)*Seattle, WA, USA.

Gibbs, G. (1994) *Improving student learning:through assessment and evaluation,* Oxford Staff Dev, Oxford.

Hartley, P. (1997) *Group communication,* Routledge, London.

Hendson, B. (1997) *Br J of educational technology*, **28, 4,** 257 - 270.

Hill, J. R. and Raven, A. (2000), Vol. 2002 ITForum.

Jaques, D. (1984) *Learning in Groups,* Kogan Page, London.

Kolb, D. (1984) *Experiential learning,* Prentice Hall, London.

Lawther, P. and Walker, D. (2001) *Education and Training,* **43,** 105-116.

Lewis, R. (2002) *International Review of Research in Open and Distance Learning,* **2**.

Lieberman, H. (1997) In *Proceedings of the ACM Conference on Computers and Human Interface, CHI-97*Atlanta, Georgia.

Lipnack, J. and Stamps, J. (2000) *Virtual teams: people working across boundaries with technology,* John Wiley, New York.

Logic Programming Associates (2000) LPA, London.

Nijholt, A. (2001) In *First International Workshop on Agents and Internet Learning*International Conference on Autonomous Agents, Montreal.

Opie, A. (2000) *Thinking teams/ thinking clients: knowledge-based teamwork,* Columbia University Press, New York.

O'Sullivan, T., Rice, J. and Saunders, C. (1996) *Successful group work,* Kogan Page, London.

Salmon, G. (2000) *E-moderating,* Kogan Page, London.

Sharan, S. (1990) *Collaborative learning: theory and research,* Praeger Publishers, New York.

Soller, A. and P. Busetta "An intelligent agent architecture for faciltating knowledge sharing communication."

Tiwari and Holtham, C. (1998) In *ITiCSE*ACM.

Tuckman, B. W. (1965) *Psychological Bulletin,* **63,** 384-399.

Vassileva, J. and Deters, R. (2001) In *First International Workshop on Agents and Internet learning*International Conference on Autonomous Agents, Montreal.

Vliem, M. (1998) University of Twente., Twente.

Whatley, J., Beer, M., Staniford, G. and Scown, P. (1999a) In *HCI International 1999*Lawrence Erlbaum, London., Munich, Germany.

Whatley, J., Staniford, G., Beer, M. and Scown, P. (1999b) *Journal of Interactive Learning Research,* **10,** 361-373.

Wooldridge, M. J., N (1995) *Towards a theory of cooperative problem solving.*