Achieving Virtual Teamwork Using Software Agents

Janice Whatley
University of Salford
J.E.Whatley@salford.ac.uk

ABSTRACT
In this paper a software agent system is described to help give online support to students who are taking part in team projects. A prototype agent system has been designed to help with allocating tasks to students in the team and to assist agreement on ground rules for working together. After two iterations of implementing this agent system with student teams, the findings from student feedback is given, with suggestions for enhancing the functions of the agent system in future prototypes.

Keywords
Team working, software agents, online learning, virtual teams

INTRODUCTION
In business the virtual team is now a reality, many companies use networks to achieve results from team working (Lipnack and Stamps, 2000). In higher education we should now give students the opportunity to experience virtual team working, so that they are more used to differences between virtual and face to face team working, and can adapt to the emerging environment at work. Undergraduate students would like to learn about team working, an important asset to employers, especially as increasingly global team working is becoming established practice (Paul, Seetharaman et al. 2004). If we can provide a non-threatening environment for students to learn these skills, then students and employers benefit.

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). Current research into e-learning is suggesting many ways in which technology can empower students, and designing information systems to help with learning activities is providing challenges.

In this paper the difficulties students encounter when working in teams are presented, with a possible solution to these difficulties, in the form of a software agent system. A prototype agent system is described, and the results from two cycles of trying this system in the learning environment are given.

LEARNING ABOUT VIRTUAL TEAMWORK
Traditional campus based programmes often incorporate team or group working as an important means of developing interpersonal skills, encouraging self-directed learning, and allowing peer learning to take place. As more teaching units are offered online, we should allow online students to gain similar team or group experiences, but with the additional notion of experience in virtual team working.

There are difficulties in providing online tutorial support mechanisms for students, and a particular problem is how students online can gain the same learning experiences as traditional campus based students (Thomas et al., 1998). In the absence of face to face communication, we need to find out about online interaction, and how the tools available may be harnessed for effective communication, in order to achieve an element of collaboration as well as co-operation, necessary for working in a team.

GROUP DYNAMICS
The processes of a group are centred around three areas: achieving the task, building and maintaining the group and developing the individual (Adair 1986). Many researchers have identified “group dynamics” as a significant factor in the processes, representing the changes in the group over time, with interacting forces that manifest themselves in the various positive and negative features of group working (Bion 1961; Cohen 1994; Johnson and Johnson 1997). Simulating the face to face environment online is not trivial, the traditional cues are missing

from communications. Computer mediated communication (CMC) are used, but offer limited support for the maintenance areas of team working.

Belbin provides us with distinctions between a team and a group (Belbin 2000). Groups comprise 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. In the context of higher education, a team is the preferred term for several students working collaboratively.

Cohesion describes a team that seems to be sticking together, and within which members wish to remain (Brown 2000). One view of team cohesion is the extent to which a team culture has been developed (Aranda et al. 1998). Team culture refers to attitudes, assumptions, interactions and the ways the team operates. These are developed as a result of common values of what is desirable behaviour within the team. Commitment to achieving the goals is high in a team of students working towards a common assessment, accountability is taking responsibility for part of the work, when tasks have been allocated, and trust is where team members can rely on other members either to complete what is expected of them or to provide a safe environment for discussion.

In learning activities where we expect students to work together in a team, the tutor has to provide support for the task, maintenance and individual development areas. Increasing student numbers are beginning to make the scale of involvement of tutors excessive, especially as even campus students are choosing to work online. Harnessing technology to provide additional support appears to be a fruitful avenue of research.

USING TECHNOLOGY TO SUPPORT TEAM WORKING

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).

APPLYING SOFTWARE AGENTS TO LEARNING SITUATION

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 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 a case study 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 (Baggetun, Dolonen et al. 2002).

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. Software agents may continuously operate in the background on a student's workstation and act autonomously to suggest ways in which the learner might improve performance.

Software Agents offer a possible source of support for learners, derived from Artificial Intelligence, they have evolved from intelligent tutoring systems to provide support to students. In this paper a software agent implementation is described, which has been designed to provide support in the form of helping students to allocate project tasks to team members, and agree ground rules for the team to follow. Both of these were found to be important elements of developing trust and group cohesion, upon which successful team working is based (Whatley, Staniford et al. 1999).

DEVELOPING A SOFTWARE AGENT SYSTEM

In this paper an application of a software agent for supporting students working on team projects 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 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 help all students to be kept aware of the status of the project.

Research methods used

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 the 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.

To see how technology can be applied to team projects, it was necessary to first 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 teams working in the face to face situation, we were 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 with some identified factors in given in Table 1.

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 – Working stages of a team project

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, Rice et al. 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.

The Guardian Agent was tested with several teams of undergraduate students on two successive academic years. These students were working on projects in systems development as part of their undergraduate programme. Teams consisted of between 8 and 14 members, working on campus. They were asked to use the allocation of tasks and ground rules functions as they began their projects, and as each team project is slightly different, the tasks were made 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 two to three 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. Team leaders were also interviewed to gain a deeper insight into the ways in which they organized their team members into different task areas.

Implementing the software agents

The first agent prototype was developed to perform a limited set of functions to help students to get started on their teamwork, i.e. allocating tasks to students according to their abilities and preferences. It was programmed in Prolog, making use of the rules and message passing features of the language.

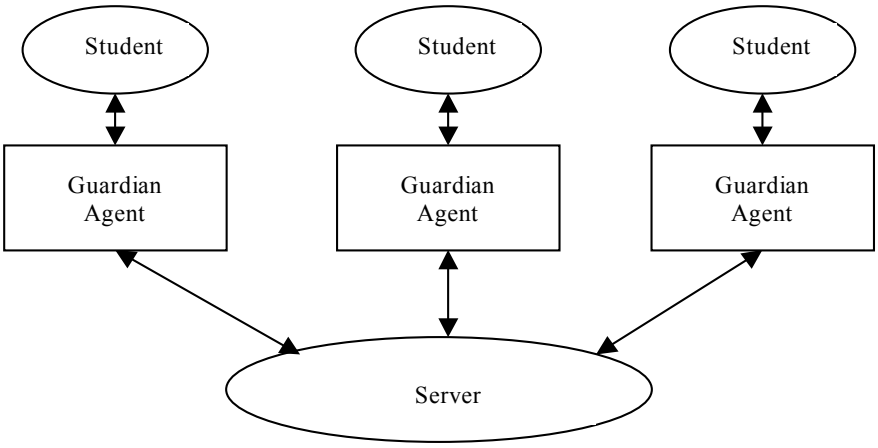

Figure 1 – Interactions between Student, Agent and Server

In the chosen system structure, each student communicates with the agent system by means of their individual Guardian 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.

The Guardian Agent will obtain its own student's abilities and preferences, asking the student to identify from a list of task areas those the student is good at, not very good at, likes and dislikes. These are posted to the server agent as facts in a database, 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 present the allocations, so that the student can consider and discuss them with the other students on the project.

The issue of agreeing ground rules for team working has been little explored (Bos, Olsen et al. 2002), and previous work had identified problems, such as difficulties getting students to attend meetings, inform the team leader if they cannot attend and complete their assigned work on time. Hence the second prototype included an additional function, to help the team members to agree ground rules for working together. Each student is asked to select from a list the ground rules they think are important, applying rules, the agent will suggest the ground rules that everyone agrees with, and those that at least half agree with. As a result of feedback the interface was altered to improve the usability, and the system was implemented in Java, using a client-server type architecture.

The results from two cycles of trials of the agent system are discussed in the next section. Prototyping enabled us to continually refine the design of the interface, and student feedback has helped us to understand some of the processes of team working that may be pertinent to working in virtual teams.

RESULTS FROM THE TWO CYCLE OF IMPLEMENTATION

Designing and implementing an agent system requires consideration of a number of issues, such as user acceptance, reliability of the functions and technical feasibility. The purposes of testing the two prototype systems were to:

- determine whether an agent system might be acceptable to students working on campus or online,
- identify processes of team working for campus students, which may be similar for online students, and evaluate the usefulness of the functions included,

- recognise usability issues that need to be addressed.

Analysis of the questionnaires showed that students felt that the agent system would be of benefit both to campus-based and online students, though when asked whether they would personally like to use the agent, the results were not as positive from students used to face to face interaction. This may be because they were comparing with their previous experience, where without an agent the teams had been used to developing coping strategies for group dynamics problems. A feeling of "big brother" was intimated through the focus group, as shown by a student comment:
"Assumes that there would be no rebellion against the agent – would not argue with the machine, simply would not do it."

Findings from the focus group indicated differences in the ways the teams have traditionally worked, from the ways in which online teams might work. For example students quoted:

"Task allocation affected by motivation, allocate tasks using a risk analysis approach – don't allocate key tasks to high risk people"

"Need to take personality into account"

More than half of the students felt that the functions included were useful, some team leaders compared the agent's suggestions with their own usual means of allocating tasks, and thought that it gave another set of data to consider, which helped with decision making. Students emphasized that other functions would help them to work together, such as:

"help with the task of communication, cope with absenteeism, help when not knowing what to do exactly…"

In evaluating the usefulness of the first prototype, issues concerning the scalability of the system, integration of the system into a user interface and portability of the system to other platforms arose. As the prototypes were being used, it became apparent that not all students were accessing the server agent at the same time, so it was difficult to determine whether a larger scale implementation would be successful. The first 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, but this was alleviated with the second prototype. However, the speed of message passing between the Guardian Agents and the server was acceptable, using the internal network with both versions.

Several students suggested that a more user friendly interface was needed, including brief instructions on screen, and making the selection of task areas more intuitive, to aid the less computer literate user. Accessibility is to be taken into consideration in the next prototype. Students identified difficulties with the present ruleset, indicating that more complex rules are needed, and there is often going to be a need for negotiation based on the suggestions:

"Issue of avoidance – if 6 people all pick same task .. delegate to person with more skills or aptitude. Make it more focused so that can allocate specific tasks rather than generic tasks, how long will task take, how many people will be needed to do task"

A student's perception of their ability might be misleading:

"Problem that it is what each individual team member <u>thinks</u> they are good at, not what their aptitude is – team project work is an opportunity to learn new things, not just about what you can do and what you think you can do."

It appears that a blackboard architecture is suitable for this type of application, offering a central repository for storing data, which can be easily accessed by any of the students' agents. An issue to consider is whether data personal to a student should be stored in the central repository, or with the agent, i.e. on the student's workstation. The latter solution may be more appropriate for online student than for campus students.

Issues arising from the trials

Student teams based on campus are as likely to benefit from additional support suggested by an agent system as are online teams. The agent system would not be a substitute for team negotiation, but would provide suggestions and advice when needed. Tutors may not always be available for providing help, so the agent system would be programmed to provide help, such as advice when team dynamics are breaking down. This advice can be built up into the knowledge base in the same way that frequently asked questions are. Similarly, shortfalls in skills may be identified and tutoring recommended to the student. An added benefit of the system was that team leaders had to select from a list the tasks relevant to their particular project, this forced an element

of thought about the structure of the project, and what was involved. Similarly, team leaders had not previously routinely considered the ground rules their team should work to.

Monitoring the project against the defined plan is an important function for the agent. The agent would identify potential problems, such as lack of communication, late delivery of work tasks and additional training needs, by comparing online message passing with the allocations, ground rules and plan stored in the knowledge base. Every time a student returns to their workstation, the agent will access the database to determine the current status of the project. A message to the student will confirm the status, and if a deadline has been missed, or a team member has been inactive, a message to this effect will be posted to the student. The agent can give advice in the form of a suggestion, broadcast to all students, who can choose whether to accept or reject this advice. Learning in the form of personal development would benefit from reflection on actions taken or not taken during the course of the project. Whether the responses from the agent should be made available only to the student involved or to the whole team, and to the tutor, is a matter for debate and further study into the impact on the team processes as a whole.

The allocation of tasks is not necessarily decided from simple rules, but more complex issues are considered. These may not be apparent online, so our agent has the added need to obtain more information in order to make suggestions. Although the software agent in its current implementation is simple, the intention is to incorporate further functionality into the system to enable fuzzy logic and data mining to be utilised within more complex rules. In this case it is essential to retain the agent character of the architecture, so that facts relating to the project are stored in a central database, but facts relating to the individual student will be stored separately. These facts will be used to monitor the learning of the student, and if an agent system were to be used over the period of an undergraduate programme, the agent will learn about its student, to be more proactive in its responses to changes in the learning environment, for the benefit of its student.

As more intelligent functions are programmed into the agent system, it may be more appropriate to use the Prolog language. However, our trials have shown that Java is suitable for coding the interface aspects of the agent system, so it appears a mixture of the two programming languages may be the best solution for a complex system.

CONCLUSIONS

In this work we have successfully implemented a multi-agent system, across an existing network, to demonstrate the capabilities of such a system. In implementing a multi-agent system one has to compromise between a complex system with many functions that will be memory intensive and require large messages to be passed between agents, and a simpler system with limited functionality where smaller messages can be passed quickly across a network, which will be usable by students with slow online connections.

We have identified a variety of difficulties students experience when working in teams, and have developed a software agent system to help students to deal with these difficulties. Students are shown to be willing to use an agent system to help them with their project work, and have suggested additional functions, which we should incorporate in future prototypes.

In future developments of the prototype the additional functions that will be incorporated into the agent system, include a monitoring facility to identify potential difficulties and suggest ways in which to avoid these becoming a problem to the project progress. In particular we would like to test the agent system on teams working solely online, in order to ascertain future functions needed for this environment.

ACKNOWLEDGMENTS

This work has been supported by an internal Teaching and Learning Quality Improvement Funding Scheme, at the University of Salford.

REFERENCES

Adair, J. (1986). Effective teambuilding: how to make a winning team. London, Pan Books.

Aranda, E., L. Aranda, et al. (1998). Teams: structure, process, culture and politics. London, Prentice Hall.

Aylett, R. (2001). Intelligent virtual agents. IVA2001, London, Springer Verlag.

Baggetun, R., J. Dolonen, et al. (2002). "Designing pedagogical agents for collaborative telelearning scenarios."

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.

Bos, N., J. Olsen, et al. (2002). Effects of Four Computer-Mediated Communications Channels on Trust Development. CHI, Minneapolis, USA.

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

Brown, R. (2000). Group processes. Oxford, Blackwell Publishers Ltd.

Cohen, E. (1994). Designing Groupwork. New York, Teachers College Pr.

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. and M. Yazdani (1998). Computer supported co-operative learning in a 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.

Hendson, B. (1997). "Groupwork with multi-media in maths: the role of the technology and teacher." Br J of educational technology 28, 4: 257 - 270.

Johnson, D. and F. Johnson (1997). Joining together: group theory and group skills. Needham heights, Allyn & Bacon.

Lawther, P. and D. Walker (2001). "An Evaluation of a Distributed learning System." Education and Training 43(2): 105-116.

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.

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

O'Sullivan, T., J. Rice, et al. (1996). Successful group work. London, Kogan Page.

Paul, S., P. Seetharaman, et al. (2004). "Impact of heterogeneity and collaborative conflict management style on the performance of synchronous global virtual teams." Information and Management 41: 303-321.

Vassileva, J. and R. Deters (2001). Lessons from deploying I-Help. First International Workshop on Agents and Internet learning, Montreal, International Conference on Autonomous Agents.

Whatley, J., M. Beer, et al. (1999). Group Project Support Agents for helping students work online. HCI International 1999, Munich, Germany, Lawrence Erlbaum, London.

Whatley, J., G. Staniford, et al. (1999). "Intelligent agents to support students working in groups online." Journal of Interactive Learning Research 10(3/4): 361-373.