Extract from:
Education in a Changing Environment 17th-18th September 2003
Conference Proceedings

Software Systems to Support Students Working on Team Projects

Janice Whatley, j.e.whatley@salford.ac.uk,
Elaine Ferneley, e.ferneley@salford.ac.uk,
Mark Jones, m.c.jones@salford.ac.uk

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. Team projects form an important part of the learning process for students in the Information Systems Institute. The particular problems of working on projects in teams are explored, and taking an action research approach, a 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, bringing together intelligent reasoning and communication across networks. The system has been tested on student teams, and the results discussed in terms of modifications for a future prototype

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), particularly if we want to include team working as one of the learning activities.

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 and on campus. In particular software agents may offer a solution to problems of supporting students undertaking team projects.

In this paper an agent system for supporting the maintenance tasks of team projects is described. 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. The structure of a prototype system to support the planning stages of a group project is described, based upon research into problems students experience when carrying out team projects in the face to face situation.

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

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

> 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. Internet searchbots (Lieberman 1997). There are several examples of software agents acting as Internet searchbots, such as Phibot or MySpiders, some are designed to 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).

E-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 (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.

In the field of e-learning software agents have the potential to help online learners in several ways. One such way is to share 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). 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).

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

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.

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, abstract 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 practise 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 comparible to campus based.

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, that is 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). 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

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

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 the Information Systems Institute (ISI), teams allocated to projects span all three teaching years on the undergraduate programmes, so first, second and final year students work together. This enables the students to learn a wider variety of skills from each other, and gives them the opportunity to discuss the project with students from other years. The projects they work on are 'real life', provided by outside clients from various local organisations. The essence of learning how to work in a team is an important aspect of team projects. Organisations make much use of team working, whether face to face, or increasingly, in a virtual team. The experience we provide is important, but is aimed at face to face teams. There is an increasing need to offer the opportunity to work virtually as well.

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 enable all students to be aware of the status of the project.

An agent prototype has been developed to perform 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, and we will describe work in progress on a client-server system for this preparation and supporting role, which will be 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, the student groups participating in the design process are broadly similar and representative, 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 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 and some identified factors are given in Table 1.

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

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 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 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 First Prototype Agent System

The initial prototype for the Guardian Agent 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.

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

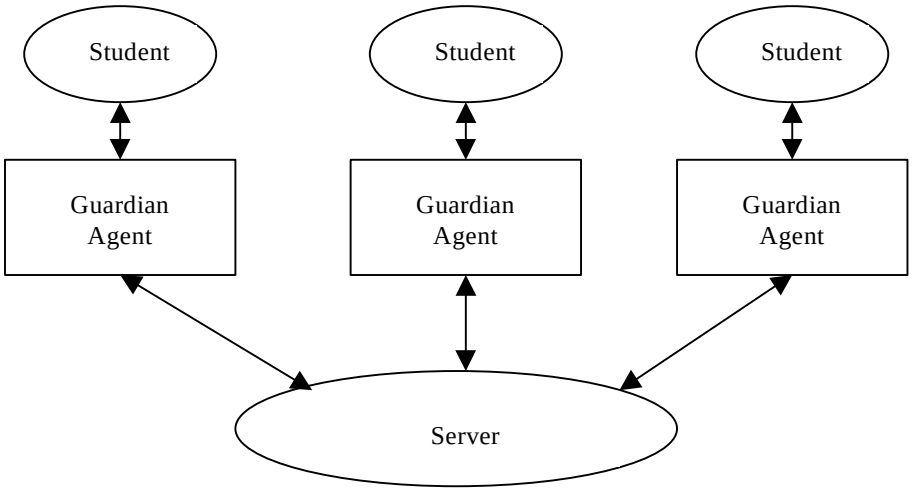

Figure 1: Interactions between Student, Agent and Server

In the chosen system structure, each student in the team, 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.

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, that is, on the student's work station. The latter solution may be more appropriate for online students than for campus students.

The server agent functions as a database, facts stored either as Prolog facts or fields in a database (D'Inverno and Luck 2001). The intelligent features of the agent are programmed within the guardian agent residing on each student's work station. Agent actions are triggered by changes in the status of the project, recognised by data changes on the blackboard or local changes within the student workspace. As such this agent implementation does meet the definition of an agent given by Ferber as:

> *…autonomous agents functioning in parallel and attempting to achieve a goal… (Ferber 1999).*

The Student/Agent Interface

The process of allocating roles begins with finding out about each other's abilities and preferences. Online there may be several students working on the team project, 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

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

agent asks its student to identify the predetermined task areas he or she likes, is good at, dislikes and is not good at. For campus based students working on a team project, one difficulty is getting all of the team members to attend at the same time to discuss preferences. The agent system would help to synchronise the collection of preferences.

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 present the allocations, so that the student can consider and discuss them with the other students on the project.

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

Allocation 1
If student A likes X and is able at X
Then student A should do X;

Allocation 2
If student B is good at X and has not expressed a dislike of X
Then student B could do X;

Allocation of tutoring
If student C likes X, but is unable at X
Then student C could be offered tutoring in X.

These basic rules were derived from the analysis of team member abilities that team leaders usually perform at present. The rules may be changed by the tutor, according to the learning outcomes desired from the project, for instance, a tutor may require students to attempt a task they have not previously done, or are not very good at. Additional rules may be used, particularly if the task areas are not familiar, for example students could be asked if they think they would like to try a task, and allocation could be based on aptitude at a similar task.

The intention is that the agent system will not replace negotiation between team members, but will offer suggestions that the team members can use to initiate discussion. This will certainly be necessary in cases where too many students are allocated to one task, or no students are allocated.

Results from the First Trial of the Agent System

The Guardian Agent was tested with seven teams in the ISI, 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

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

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. A summary of the results from the questions asked is given in Table 2. Students who did not answer 'yes' were roughly equally split between those who did not answer the question and those who answered 'no'.

The interface was generally acceptable (89%) but some students suggested improvements, which were incorporated into the next prototype. About half of the students said that the output from the allocation of tasks function was useful (56%). 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 (81%) as well as for campus based students (64%). Just over half of the students said that they personally would like to use such an agent (56%).

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

Table 2: Results of the Questionnaires Completed by Students

But, interestingly, few students said they would like to see a character representation of the agent (16%), the Microsoft PaperClip agent does not appear to be very popular.

These results prompted us to pursue this avenue of research, as there is clearly some benefit to the students on campus. Although the questionnaires gave us some quantitative results, these are not significant in themselves, but give a general feeling for the students' opinions. Of more importance is the answers given to open questions and the comments received in the focus group, which give us richer data upon which we can further develop the system.

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

Discussion of the Results

The purposes of the two prototype systems were to:

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

Designing and implementing an agent system requires consideration of a number of issues, such as user acceptance, reliability of the functions and technical feasibility. These will be discussed in the remainder of this section, illustrated with student comments.

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. This may be because they did not recognise its usefulness, also comparing their previous experience, where without an agent the teams had developed coping strategies for group dynamics problems. A feeling of 'big brother' was intimated through the focus group:

> *Assumes that there would be no rebellion against the agent – would not argue with the machine, simply would not do it.*

Some of the findings from the focus group showed differences in the ways the teams in the ISI have traditionally worked, and the ways in which online teams might work. For example:

> *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.*

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 were considered. The first prototype was used by seven teams, about 55 students in all, but 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. However, the speed of message passing between the Guardian Agents and the server agent was acceptable, using the internal network.

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. Modifications to enhance the functionality were suggested, such as:

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

> *Issue of avoidance – if 6 people all pick same task … delegate to person with more skills or aptitude. Make it more focussed 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 thinks they are good at, not what their aptitude is – team project work is an opportunity to learn re new things, not just about what you can do and what you think you can do.*

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 it was decided that an additional functionality should be included, to help the students to agree ground rules for the process of working together.

Taking into consideration the feedback from students and issues of portability, we decided to build the second prototype in Java. As more intelligent functions are programmed into the agent system, it may be more appropriate to use the Prolog language. However, 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, albeit a simple one, 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 between agents.

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 accepting of an agent system to help them with their project work, and have suggested additional functions, which we should incorporate in future prototypes.

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.

Acknowledgements

This project was made possible by funding from the University of Salford Teaching and Learning Quality Improvement Scheme and by the Vice-Chancellor's Scholarship Fund.

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

Further Information

For further information about the initial or second prototype please contact Janice Whatley, J.E.Whatley@salford.ac.uk.

References

Aylett, R. (2001). *Intelligent virtual agents*. IVA2001, London, Springer Verlag.
Baggetun, R., J. Dolonen, et al. (2002). "Designing pedagogical agents for collaborative telelearning scenarios."
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.
Connolly, J. (1994). *CSCW and artificial intelligence*. London, Springer Verlag.
D'Inverno, M. and M. Luck (2001). *Understanding agent systems*. Berlin, 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 cooperative learning in a virtual university. Internet, Virtual University.
Ferber, J. (1999). *Multi-agent systems*. Harlow, Addison Wesley
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 Centre for Staff Development.
Hartley, P. (1997). *Group communication*. London, Routledge.
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.
Henninger (2002). Phibot. 2002.
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, LPA.
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.

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/

Pant, G. and F. M. E. y. o. i. W. c. A. A. a. M.-A. S.-.-. Menczer, 2002 (2002). "MySpiders: Evolve your own intelligent Web crawlers." *Autonomous Agents and Multi-Agent Systems* 5(2): 221--229.
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
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, Kogn Page.
Thomas, P., L. Carswell, et al. (1998). "A holistic approach to supporting distance learning using the internet: transformation, not translation." *Br J of educational technology* 29, 2: 149 - 161.
Tiwari and C. Holtham (1998). *Learning Groupware through using groupware - computer supported collaborative learning with face to face students*. ITiCSE, ACM.
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.
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*.

ISBN 0902896660

Published by the University of Salford
http://www.edu.salford.ac.uk/her/