Investigation into the Roles of Agents in Supporting Students Working on Group Projects Online

Janice Whatley

University of Salford

J.E.Whatley@salford.ac.uk

Martin Beer

Sheffield Hallam University

M.Beer@shu.ac.uk

Abstract

Group projects give students an opportunity to discuss their understanding of the subject with their peers, as they apply the theory to practice. We should try to incorporate an element of groupwork in online learning and teaching provision, but it is difficult to simulate some of the processes of groupwork online. In this paper we present initial work on developing a software agent which can help in supporting students working online on group projects. This agent is to be incorporated into a learning environment being used to support the OTIS learning and teaching provision. The implementation is described together with its potential for enhancing the support given to online learners.

Keywords

Software agent, group projects, online learning

Introduction

Group projects give students an opportunity to discuss their understanding of the subject with their peers, as they apply the theory to practice. However, the dynamics of group working are complex, and not supported readily by present communication tools. The maintenance as well as the task aims of the group project need to be considered. A software agent, called a Guardian agent is suggested, which can be embedded into an environment to give additional support for some of the maintenance aims of group working. The OTIS (Occupational Therapy Internet School) learning environment is used as a medium for providing the agent support.

Group projects and online learning

Successful group working requires that the maintenance roles as well as the task roles of the group 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 group develops (Bion, 1961; Gibbs, 1995; Jaques, 1984). Traditional undergraduate campus-based courses incorporate a group project element, as the essential means of "learning by doing", but to accomplish a group project online requires different skills and ways of working. In this work we are looking at the potential for intelligent agent technology to help students working online to acquire the skills and ability to collaborate successfully on group projects.

A number of problems arise when adapting a conventional approach to group projects for students working online:

> Students are working at their own pace and in their own time, so a timetable must be imposed either by the group itself or by the course leader;
>
> Organising conventional meetings is not possible, but a substitute is necessary, probably aided by technology;
>
> Sharing information must be enabled by technology, students must be able to express their opinions online, which may require different skills;
>
> Assessment is probably not possible on an individual basis, but a group mark may not be acceptable if students recognise that members of a group are not pulling their weight;
>
> Tutors may experience difficulties monitoring the progress of groups of students.

Computer mediated communication (CMC) tools, such as conferencing, email, discussion forums support the communication needs for the task roles of group projects, examples include studies of co-operative learning in a virtual university (English & Yazdani, 1998) and groupwork in mathematics teaching (Hendson, 1997). Student support using commercial groupware products enables communication between group members and instructors. BSCW has been used as support for group projects and was found useful for information sharing, offering greater flexibility in students' face to face communication, but offered limited support for the maintenance roles of groupwork (Vliem, 1998).

Managed learning environments (MLE) are being used for a variety of purposes for online learning. These incorporate different forms of computer mediated communication (CMC) to assist with some of the difficulties listed above. However, what is missing is the extra advice to students to help them to decide which of the many features are appropriate at any particular time. The MLE's provide a structure to enable communication, but little help in the process of communication to help form learning networks (Lawther & Walker, 2001). The alternative, or additional, approach suggested is that MLE's should have an extra layer of intelligent help incorporated to monitor the individual students and give individualised advice as required.

Problems encountered in group working

Students undertaking group projects in a traditional setting face several problems to a greater or lesser degree:

> the time taken for a discussion and to reach collective agreement;
>
> the time it might take to recognise potential problems;
>
> getting all members to agree their responsibilities;
>
> knowing who can do which parts of the project;
>
> recognising when extra help with skills is needed;
>
> bringing the project together.

The results of surveys on group projects have been considered closely when deciding what functions software agents should undertake in supporting group working (Whatley et al, 1999). Our work has included research into typical problems students encounter when doing group projects in the face to face context, together with a pilot implementation of a Managed Learning Environment for learning occupational therapy (Armitt et al., 2001). Work was undertaken as part of this study to identify the most effective ways that students learnt within the group environment (Armitt et al, 2002). These studies were used to identify limitations in the operation of current tools for learning environments and resulted in an analysis of the stages of a group project where students could be guided to make most effective use of the learning opportunity offered. We then constucted a prototype system of intelligent agent support for one of the major areas identified.

Guardian agent solution

To see how new technology can be applied to group projects, it is necessary to analyse the stages of a group project, to determine the particular problems encountered at the different stages, and to determine whether new ways of working may be possible, rather than simply adapting a current traditional project, (Table 1).

Table 1 – Roles identified for the Guardian agent at each stage of the group project.

Project stage	Potential areas in which Guardian agent can help
Planning	Introductions Setting ground rules Produce a project plan
Doing the project	Check the time schedule Ensure all members contribute Identify lack of skills
Completing	Collating the individual parts Preparing a report Appraising the group's performance

It is proposed that a Guardian agent can offer some support to students, working quietly in the background on each group member's workstation. The agent will autonomously monitor the progress of the group project, suggest ways in which the students can act to improve the progress of the project and enhance the communication between members of the group. Each student working on the project will have an individual agent, and agents belonging to each member of the group are able to communicate with each other to exchange information and negotiate on behalf of the students. This agent does not replace the tutor's input, but performs some of the administrative tasks, which are usually performed by the group members during face-to-face meetings automatically. This allows the tutor to concentrate on problem-solving and academic issues, rather than the simply organising the group, as is often the case in web based tutorials (Armitt et al., 2002).

In our architecture each individual agent will have a similar structure when the project begins. Each agent will have 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 (Figure 1).

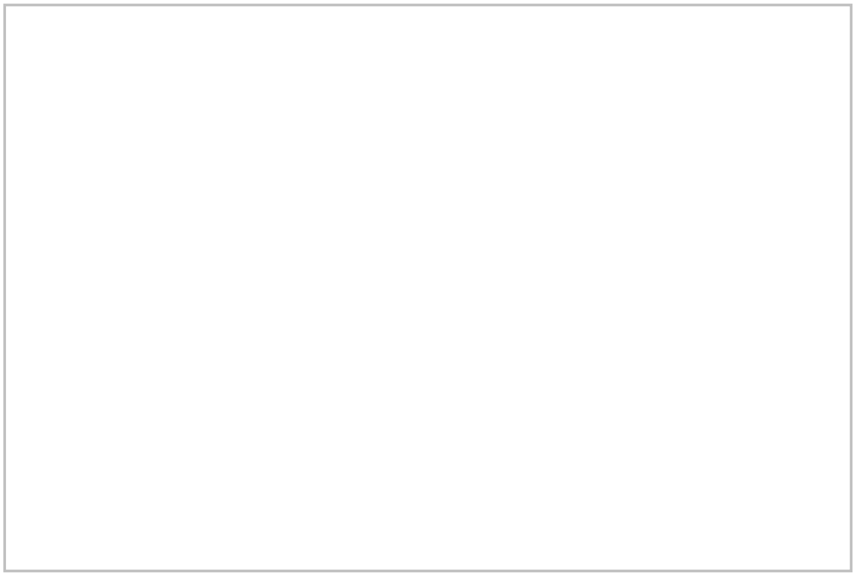

Figure 1 Architecture of Guardian Agent System

Prototype developed

The initial prototype for the Guardian Agent is being developed in LPA Prolog, using their Agent Development Kit (LPA, 2000). This tool enables the developer to code the interfacing aspects of the agent without worrying about the technicalities of the agent communication, which are 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.

There next follows a description of the processes we went through for designing the allocation of roles functionality.

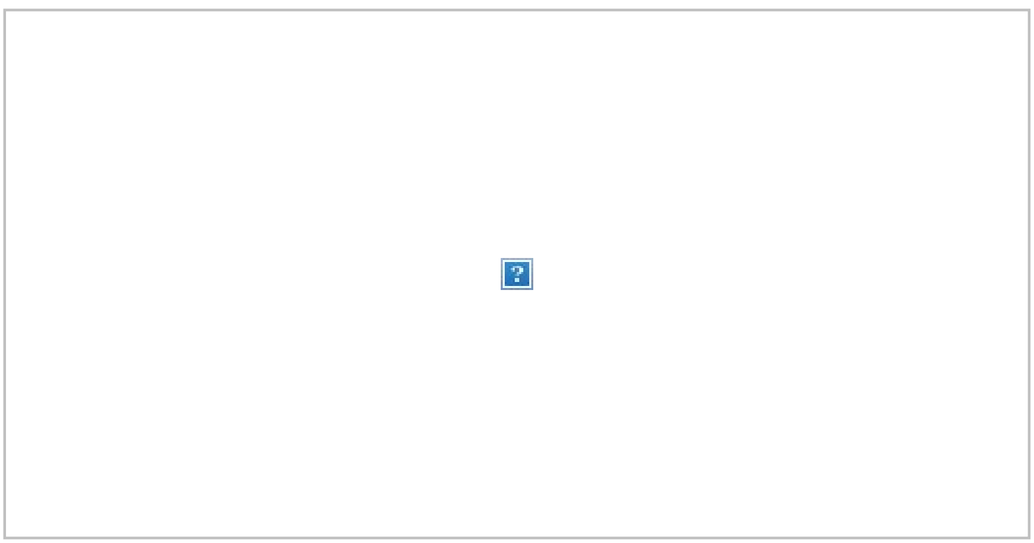

Figure 2 – Use case diagram: Introductions

Using object oriented modelling the use case diagram shows the different parts of the code that will have to be written, according to the possible ways the student might interact with the Guardian agent (Figure 2). The use cases have been expanded textually to indicate the sequence of events that needed to be coded, for example the use case "Check all students responded" is given in Table 3.

Use Case Introductions B Check all students responded

1 Agent reads posted facts from server agent

2 Agent compares posted list with student list

If lists match:

3 Then inform student and

4 Agent finds allocations

Else:

5 Inform student that we are waiting for other students

Table 3 – Expanded use case for checking whether all students have posted

The process of allocating roles begins with finding out about each other's abilities, so given that there will be several students working on the group project, each accessing the project site at different times and not knowing which of the other students have already started using the site, the first function of the Guardian agent is to determine whether or not its student has already posted their abilities. Where the agent finds that its student has not posted their abilities, the agent asks its student to identify the predetermined task areas he or she likes, is good at, dislikes and is not good at.

This process continues for each of the students and their Guardian agents, until all students have posted their abilities. At this point the Guardian agent accesses the facts stored on the Server agent and, using a series of task allocation rules, determines which students might like to be allocated to carry out which tasks. These allocations are posted to the Server agent, where each Guardian agent can read them off and present them to their student.

We have programmed the Guardian agent to work with three levels of allocation: allocation1, allocation of tutoring, allocation2, which use 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 training in X

Allocation2 –

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

Then studentC could do X

Students working on the group project will have at their disposal a variety of CMC tools, including email and discussion forums. It is likely that there will be some conflict to resolve, such as too many students being allocated to the same task, or no student allocated or offered training for a task. After informing the students of the conflict, the students would be encouraged to discuss the allocations given, and agree to accept them or negotiate alternative allocations. 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.

Next the students have to agree a project plan. Once again the Guardian agent prompts its student to suggest tasks that need to be performed and the time they will take. 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. 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.

Our demonstration will be a basic prototype guardian agent that is currently being evaluated to prove that it can significantly enhance the effectiveness of web-based group project work in the ways intended, and to evaluate 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.

INTEGRATING WITH ONLINE COURSEWARE

For the Guardian agent to work effectively it must be integrated into the overall student learning environment. This section discusses how this can be done in the OTIS learning environment (Armitt et al., 2001). The OTIS system design arising from the requirements analysis identified the need for a core system which would be common to all future courses, and a course-specific element comprising the course materials. The core comprises the user and document management facilities and communications. The course materials are developed in HTML using standard Internet tools. Multimedia are developed using standard tools and 'plugged into' the HTML.

While the specifics of the course materials will be different for each new course, each course would be expected to comprise of the same general areas, such as case studies, reading materials and course guide documents. A generic skeleton structure within the HTML common to all courses was therefore envisaged, which would also provide the integration with the underlying core system. This allows the Guardian Agent Architecture to be integrated into the 'core' so that new agents need not be developed for each course.

Architecture

The basic architecture of the OTIS system is shown in figure 3. Both the core software and the user interface are based on a Virtual Campus metaphor (Gibbs et al., 1999) that comprises a number of virtual rooms such as the library, student work area, help desk etc.. This provides three advantages:

- more intuitive navigation around the system
- each virtual room engages the users currently in the room in a communications group
- the Virtual Campus provides a generic layer in the user interface which will be common to all courses

Figure 3 - Basic architecture of the OTIS system

OTIS Course Documents and Multimedia **(course-specific)**	*)* *)* *)* *) User* *) Interface* *)* *)* *)*
Virtual Rooms Environment **(same for every course)**	
Interface between Core and User Interface **(same for each course)**	*) Core* *) Interface*
Central Core, ***Collaborative working environment*** **(same for each course)**	*)* *) Core* *) Software* *)*

Web server software	***) Web***
Systems and backup software	***) Server***
Hardware	***) (basic system*** ***) required for each*** ***) course)***

Core software

The core software provides communication and document management facilities, as well as handling user accounts and providing secure logins to the website. The CoMentor software package, developed by the University of Huddersfield (Gibbs et al, 1999), forms the basis for the core. CoMentor is based on a multi-user object oriented database, and uses the LambdaMOO (Curtis, 1993) technology used for distributed multi-player adventure games.

Within the OTIS system, 'player' (student) objects have a property 'location' (virtual room) describing their current position in the virtual world. Players can 'Talk' to other players at the same 'location'. This allows students and teachers to meet by entering the same location or virtual room within the learning environment. Users can also 'Page' one or more users located anywhere throughout the system. Talk and Page conversations can be recorded for personal purposes so that records could be maintained for future study and analysis. All communication is currently text based, with students and tutors typing messages to each other in the chat box. Internal OTIS email is also available.

The CoMentor core has been enhanced to provide reusable modules, which can be plugged into the user interface as required. A meeting room booking system has been developed, which auto-generates meeting rooms when the booking is made. This module is used in the OTIS Meeting Rooms and Patient Consultation Area. A notice board module provides notice boards for all to see, but notices may only be pinned up by staff, or students, as appropriate.

Layout of the User Interface

The user interface is shown in figure 4, and is based on a two-part frameset specified within the CoMentor software. This splits the screen into a left hand area containing the course materials, and a right hand area providing the facilities offered by CoMentor. Within the course materials side of the screen is a further three-part frameset specified within the HTML code, described in more detail below.

The right hand side of the screen is divided into a lower 'action' area and an upper 'reporting' area. The action area provides a box into which users type messages, before clicking on the Page or Email buttons to select the recipients, followed by clicking on the Send button. When sending a Talk message, the user needs only to click on Talk, for the message to be received by all users in the same virtual room as the user. The People button gives access to a list of users and their personal areas. The Map gives a main map that can be used to return the user to the Entrance Hall and elsewhere if a user becomes completely lost.

The reporting area displays received Talk and Page messages, confirms the content and transmission of sent messages and the current position of the user, and reports on who is entering or leaving the user's current room.

Figure 4 - User interface

Figure 5 - Virtual room

The virtual rooms metaphor in the core has been carried through to the course materials. The HTML documents comprising the course materials are laid out as a series of rooms in a virtual college. All rooms are accessed from one of two main maps, and the relationship between the rooms is shown in figure 5.

Virtual rooms environment

Room	**Functionality**
Entrance Hall	The first room encountered when logging in, it provides access to six rooms and to the Courtyard. It contains the 'official' noticeboard, which everyone can read, but on which only staff can pin up notices.
Courtyard	A large subsidiary area, providing access to a further five rooms.
Library	Room containing the case studies and assignment materials, reading material, websites, web search and library search.
Student Work Area	The student's desk, giving access to their document management area
Meeting Rooms	Meeting rooms for students, tutors and others. The meeting rooms must be booked, and are created dynamically when meetings are arranged. There is no limit to the number of meeting rooms.

Lecture Room	A lecture room, still under development, which will allow online lectures and seminars to be given using audio-conferencing.
Patient Consultation Area	Bookable rooms, created dynamically, where virtual consultations are held with patients or with tutors role-playing patients.
Exhibition Area	An area providing bookable booths, where company representatives and others can display posters and other materials for the duration of the course or for a more limited period. The representatives can define times when they will man their booths and meet staff and students there, to discuss their products and provide additional information.
Computer Help Desk	An area where users can access technical documents and online technical tutorials. Problems can be reported here, by email or directly to technical staff who may man the help desk from time to time. The area also houses a list of FAQs (Frequently Asked Questions) which is built up during the course.
Administration Area	An area containing documents pertaining to the administration of the course, e.g. the course guide and prerequisites, the timetable, assessment guidelines etc. This area will also house the student registration system.
Student Café	A student-only area facilitating informal discussions between different groupings of people. The room also contains a notice board on which students can pin their own notices.
Staff Common Room	A tutors-only area, in which staff can communicate without being overheard by students and giving access to student assignments submitted for marking.
Quiet Room	An area into which students can retreat, if they wish to indicate to other people currently logged on that they are not currently available for communication. This enables students to leave their desks for short periods without logging off.

Since individual rooms are associated with different student activities it is possible to assess current activity by tracking both position within the OTIS world and joint activity. It is important for example that collaborative activities are undertaken by the whole group, who because of the synchronicity of the communication will have to be colocated. If members are missing the Guardian agent will note this and warn both those missing and the rest of the group that their goals are unlikely to be achieved. More sophisticated analysis can be undertaken by monitoring the group discuassions (Armitt et al., 2002).

Conclusions

In this paper we have described the design of the Guardian agent and how it will support students who are working on group projects online. The agent is integrated into the OTIS system environment as a vehicle for evaluation of the prototype. The prototype demonstrates the practical feasibility of this architecture for message passing between agents in a system, for the specific domain of online learning. The subset of functionality that this prototype possesses will be sufficient for us to determine the usability of such a system for supporting students, both in terms of the interface and in terms of the acceptability of such a system for students and tutors.

References

Armitt G, Green S, & Beer M. (2001) Building a European Internet School: Developing the OTIS Learning Environment, *European Perspectives on Computer-Supported Collaborative Learning, Proceedings of the First European Conference on Computer-Supported Collaborative Learning,* March 2001, Maastricht, Netherlands, pp 67-74, Maastricht McLuhan Institute.

Armitt, G., Slack, F., Green, S., and Beer, M., (2002), "The Development of Deep Learning During a Synchronous Collaborative On-line Course", Proceedings of CSCL-2002, pp 151-159, Laurance Elbaum Associates, New Jersey, USA.

Bion, W., *Experiences in groups*. London: Tavistock Publications, 1961.

CoMentor, University of Huddersfield, http://comentor.hud.ac.uk/

Curtis P, *LambdaMOO Programmers Manual*, Xerox Park, USA, August 1993.

English, S. & Yazdani, M. (1998). Computer supported cooperative learning in a virtual university. *Virtual University Internet publication*, http://www.mbc.co.uk/virtual-university-press/vuj.

Gibbs, G. (1995). *Learning in Teams.* Oxford Centre for Staff Development.

Gibbs G, Skinner C & Teal A (1999) 'CoMentor: a collaborative learning environment on the WWW for philosophy and social theory students', http://www.hud.ac.uk/comentor

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,* Vol 28 No 4, pp257 - 270.

Jaques, D. (2000). *Learning in Groups*. London : Croom Helm.

Lawther, P. & Walker, D., (2001), "An evaluation of a distributed learning system", Education and Training, Vol 43, No 2, pp105 – 116, MCB University Press.

LPA, (2000), Logic Programming Associates,

Vliem, M. (1998). "Using the Internet in university education", Ergonomics Group, University of Twente, Netherlands.

Whatley, J., Beer, M., Staniford, G. and Scown, P. (1999). "Group Project Support Agents for helping students work online", Proceedings HCI International, Lawrence Erlbaum, London.