
Developing robotics e-teaching for teamwork

Blaine A. Price*, Mike Richards, Marian Petre,
Anthony Hirst and Jeffrey Johnson

The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK

E-mail: B.A.Price@open.ac.uk

*Corresponding author

Abstract: This paper addresses whether students can e-learn teamworking skills when most are logged on to the Internet without direct personal interactions. We give examples of how the Open University achieves this and argue both that teamwork skills can be learned through electronic interactions and that robotics can be a particularly effective focus for that experience.

Keywords: Distance learning; educational robotics; group working; teamworking.

Reference to this paper should be made as follows: Price, B.A., Richards, M., Petre, M., Hirst, A. and Johnson, J. (xxxx) 'Developing robotics e-teaching for teamwork', *Int. J. Continuing Engineering Education and Lifelong Learning*, Vol. X, No., Vol. X, No. Y, pp.000-000.

Biographical notes: Blaine Price is a Lecturer in the Department of Computing at the Open University. He holds BA and BSc degrees from Queen's University at Kingston and an MSc from the University of Toronto. He worked for Apple Computer, Inc., in California before joining the Open University in 1991. His research interests include software visualisation, computer science education and robotics.

Mike Richards is a Project Officer in the Department of Computing at the Open University. He holds BSc and MSc degrees from the University of Wales. He has worked for the Department of the Environment and the University of Wales before joining the Open University in 1996. His research interests include robotics, computer security, privacy and computer graphics. He is a frequent reviewer for IMDB.

Marian Petre is a Reader in the Department of Computing at the Open University and an EPSRC Advanced Research Fellow. She holds a BA from Swarthmore College and a PhD from University College London. Her research interests include the empirical methods for the investigation of user interfaces, design processes and complex information artefacts.

Tony Hirst is a lecturer in Artificial Intelligence in the Telematics Department at the Open University. He holds a BSc from the University of York and a PhD from the Open University. After two years postdoctoral research at the University of Portsmouth, he returned to the Open University in 1999. His research interests include evolutionary computation, intelligent software agents and robotics education.

Jeff Johnson is Professor of Complexity Science and Design and Head of Department of Design and Innovation in the Open University. He holds BA and PhD degrees from Essex University. He was a Research Fellow in Mathematics from 1971-1977 and a Senior Research Associate at Cambridge University before joining OU in 1980. His research interests are in the design, management and control of complex socio-technical systems and in the mathematical and computational foundations of complexity science.

1 Introduction

Teamworking skills are seen as essential in modern engineering education. Most projects require individuals to work with others to achieve a common aim. Members of a group must learn to manage personal interactions and frustrations while trying to achieve the project goals. These skills are not widely taught in the conventional curriculum, which has tended to concentrate on individual development. Teamworking expands an individual's skill base by requiring participants to interact and understand one another's specialisations by using each other as resources, as well as learning to work together [1]. The group learning environment is thus potentially far richer than an individual learning environment.

Professional bodies such as the British Computer Society and the Institute of Electrical Engineers increasingly require evidence of projects involving teamwork for the award of professional qualifications and recognition of degree program curricula. Teamworking is a subject that is difficult to teach solely from a book or in a classroom: students learn best from practical experience. Organising such practical activities in a conventional university environment can be challenging enough, but distance teaching, or e-teaching, of a subject that inherently involves a high degree of person-to-person interaction presents even more problems.

In this paper we propose the use of robotics topics as a problem domain for teaching teamworking at a distance. We discuss how we have used e-teaching for a robotics course and various courses involving teamwork with computing or technology. We then look at Problem-Based Learning (PBL) as an educational approach for a course teaching *both* teamwork *and* robotics. We go on to identify some of the models for designing a teamworking project and offer an outline of how we are going to use these to inform the development of our course.

Robotics is well suited as a problem domain because it intrinsically requires multidisciplinary knowledge. Robotics encompasses subjects such as mechanical engineering, electronics, control, communication, vision, real-time parallel computing and systems design. By choosing a problem domain with a high multi-disciplinary requirement, we reduce the chance that any one person on the team will 'cheat' and do all the work.

Of many examples in the literature, Beer *et al.* [2], give a clear account of how Case Western University teaches engineering and teamwork by means of robotics. Their account of student teamworking experiences is particularly compelling, as is their management of a competitive teamworking process and methods of grading their students. Students on robotics courses at conventional universities have intense face-to-face interactions with their fellow students and there is no doubt that most learn a great deal from this about themselves and about how to work with others to achieve a common goal.

By definition, this level of intense daily interaction is not possible in distance education. Regional face-to-face sessions such as tutorials and residential schools can help, but with limited impact. The Open University (OU) teaches around the world (although mainly in the UK and EU) solely through supported distance learning. Students are geographically diverse and although some face-to-face sessions are offered in most courses, fewer than half of the students attend for reasons of geography, family commitments or by virtue of their employment. For instance, the OU has a large number

of students serving with the British armed forces (including students serving on submarines underwater for months at a time).

2 Why robotics for e-learning teamwork?

As the literature supporting the positive educational effects of robotics accumulates, it becomes increasingly clear that:

- robotics is very absorbing and enjoyable for many children and adults
- robotics events and competitions motivate the study of technical subjects
- robotics provides hands on examples for teaching science, technology, engineering and mathematics (STEM) subjects
- robotics creates excellent possibilities for learning teamworking, especially through competitions.

Cumulative experience suggests that robotics is an excellent domain for introducing students to a range of technological and scientific disciplines. Robotics provides leverage on two key pedagogic aims of e-learning: problem-based learning and engaging students in discourse (i.e. creating a community of learning). Robotics possesses a number of advantages that make it particularly well suited to unsupervised project work at a distance:

- The advent of Lego Mindstorms has shown that it is possible to provide robotics kits that are small and sufficiently cheap to be economically viable, but also rich enough to be engaging.
- Lego-based robotics draws on skills derived from play and other commonplace experiences. Play is an under-appreciated part of the learning experience, giving people an excuse for open-ended exploration. We believe that play allows people to rehearse and prototype their projects without a sense of embarrassment and may well encourage less-confident students to come forward and make useful contributions.
- The end result of robot construction is concrete and observable, making it suited to independent working. By contrast, (in particular with other computer science projects), results tend not to require a new aesthetic of interpretation and appreciation (e.g., '*Gee, the robot moved toward the light*' rather than '*Oh, what a nice stack*').
- Robots are physical objects. People identify strongly with *their* team's robot and its success (or failure). The possibility of creating a tangible and exciting end product may well encourage people to work harder than if they were developing an intangible artefact such as a piece of software.
- Students get to solve lots of small problems with practical outcomes and hence experience problem resolution whose success they can also assess independently (while also experiencing larger and more abstract obstacles for which there is support).
- Students have to negotiate their final solution from the individual contributions of their group.

2.1 Teaching teamwork

Distance learners often have little opportunity for face-to-face contact, although some courses offer up to two hour-long monthly tutorials or one-off day schools. However, attendance at such tutorials is not compulsory and there are many reasons why distance education students may be unable to attend tutorial meetings.

The Open University has experience of teaching remote teamworking in both computing and technology; the two mainstays of robotics. Because of the nature of OU teaching, it makes sense to try to use electronic communications technologies to facilitate group working processes, rather than face-to-face meetings. Courses that include teamworking aim to give students a taste of working with a group of colleagues (OU students can feel very isolated in comparison to those people studying in conventional institutions).

The students are introduced to the necessary technologies of e-mail and electronic conferencing well in advance of the beginning of the teamworking and this allows them to familiarise themselves with the technology and obtain some experience of conversing with their fellows. Students are also given some training in the necessary social skills (so-called netiquette) and are taught how to work constructively, even if they are required to critique a fellow student's work.

Since 1995 we have implemented a variety of Computer Supported Cooperative Work techniques in our teaching of computing; including asynchronous electronic text conferencing, synchronous text and video conferencing and fully electronic student assignments, where tutors mark up a student document and return it to them [3–6].

At all times, e-teaching has been designed to be suitable for a *reasonably equipped* student. Many other experiments have equipped students with computers and network connections that are far in excess of anything that could be expected in the home. In contrast, we have always designed our teaching around a relatively low specification computer with a slow modem connection. As technology has improved and improvements made to internet services it has been possible to expand the scope of the electronic component of courses.

Electronic teaching has been designed to complement conventional distance learning materials. Our earliest experiments (named MZX in internal OU nomenclature) complemented two pre-existing courses (M205: *Fundamentals of Computing* and M353: *Programming and Programming Languages*) and were offered to a limited number of self-selecting students having access to e-mail. E-mail replaced postal contact and telephone calls between tutors, students and members of the course team. Students used identical material to their conventional colleagues, only the method of tuition differed. This allowed direct comparisons between the two cohorts.

The success of the first trials allowed more ambitious experiments during years two and three. These experiments comprised over 40 students spread across the UK and mainland Europe. An asynchronous conferencing system was provided on a dedicated server, providing a number of conferences for each tutor and their students. Tutors used the conferences to hold asynchronous tutorials, which might last several days. The tutor would post a problem at the beginning of the tutorial and students were invited to contribute their solutions. Tutors additionally used the conferences to post revision materials and supplemental documents of their own devising. Students could use the conferences to discuss any problems they were having with the materials.

Developing robotics e-teaching for teamwork

This second trial also invented the OU's electronic assessment system [5]. Students downloaded a copy of the assignment document from the course website and completed it with a word-processor. When they had completed the assignment, the student returned it using a form on a web page. On receipt of the assignment, the electronic assignment system automatically notified the tutor that it was ready for marking. The tutor could collect the assignment from a secure website and mark it using a modified version of Microsoft Word. The tutors returned the marked assignments via the web. The marks were automatically extracted and recorded in a database, whilst an e-mail message was dispatched to the student informing them that the marked assignment was ready for collection.

The MZX trials had been put in place to test the viability of electronic assessment and tuition in more ambitious courses that would replace M205 and M353. In 1998, M205 was replaced by M206: *Software Engineering, an Object-oriented Approach*; a course reflecting the radical changes in the software development industry. From the start, M206 had been designed to include electronic distance learning. M206 incorporated variants of the technologies developed in the MZX trials but also prototyped electronic group-working. Group-working had been a principle ambition of the M206 course developers; they believed that group-working was a key skill for anyone working in the modern software industry.

A series of group-working projects were created for M206. Each project would run for several weeks in parallel with other parts of the course and was divided into stages. Each stage was associated with a task that had to be completed before a cut-off date. A typical group-working project may run as follows. Students are split into groups of around 25 members, with each group having its own moderated conference. A problem is posted to the conference on the first day of the project. Sample problems include writing an appraisal of a simple computer interface, or developing a fragment of Smalltalk code. Students work on their solutions to the problem and post back to the conference – however, all answers are hidden until the end of the stage. Hiding solutions ensures that no student has prior sight of other answers; weaker students are not discouraged from posting by better solutions and there is less chance of plagiarism.

At the end of the stage, all answers become visible. Stage 2 is usually tasked with critiquing the work undertaken in Stage 1. Students are paired off, each either testing or critiquing the other's work – *in a constructive manner!*

M206 group-working allocates marks on an individual basis – addressing student concerns that they may be penalised for a weak group performance.

In recent years, several technology (engineering) courses have explicitly taught and assessed teamwork and group-working skills. Many Level-2 (or second year in North American terminology) technology courses (THD204: *IT and Society*; T293: *Communicating Technology*; T209: *Information and Communication Technologies: People and Interactions*) have explicitly developed communication and learning skills as well as instructing students with technical knowledge. Remote group-working facilitated by Computer Mediated Conferencing (CMC) has opened up group learning and group development activities for (often isolated) individual distance learners.

Students are placed into small groups of between four and six members from within the same tutor group. Kear [1] reports how in the first year of presentation, one course allowed students to pick their own groups, but this proved inflexible and administratively burdensome; subsequent presentations required the tutor to place their tutees into smaller

groups; and must then decide on their individual roles within the group, allocating tasks and taking personal responsibility for completing those tasks, as well as scheduling group activities.

Typically, the assessed submission (of a single, group report or set of web pages) includes a group-authored component (such as an HTML index page, or the introduction and conclusion sections of a report) as well as individual components. There are several ways that a group may approach the authoring of the group component: for example, students could pass the baton; each produce their own version of the section and then condense them to a single version; or each write a particular part of the section [1].

A formative assessment prior to the group work assessment is used to establish the group and get them to discuss ways in which they might handle various parts of the project, including task assignment and scheduling, as well as developing contingency plans for losing a member of the group. As well as explicitly asking students to address these issues (in a somewhat passive way) the formative assessment also introduces the individual group members to the realities of working in a group. It provides them with useful 'for real' experience of the group-working processes they have been asked to consider formally. At this point, however, the group members are not asked to reflect on their initial group performance.

In student surveys of one course (*Communicating Technology*), approximately one-eighth of the respondents acknowledged problems with the group activity. These problems included issues surrounding the motivation of other members of the group; the perception that assessment was unfair because an individual's mark depended on other people's effort; and problems such as working around holidays. Other difficulties include the differing commitments and priorities of the students (bearing in mind the constituency of open, distance learning) and the different pacing followed by each student in the group [1].

Kear [7] identifies several important factors that must be taken into consideration when designing the group task and its method of assessment:

- The task must be separable into group and individual components.
- The process of collaboration should be assessed. Without this, there is the danger that the group members won't address the process learning aspects of the activity [1]. She also notes how requiring a minimum pass rate of 30% on the mandatory group assessment means the group work can't be avoided.
- The group needs to be established early. A short formative group assessment to establish the group is one way of achieving this. The 'real' group assessment typically took six weeks (courses run seven months – February to October).
- The task must also catch the students' imagination and make them want to discuss it with each other in an electronic forum.
- It should also encourage critical thinking and provide scope for different kinds of interaction within the group (for example, sharing ideas/resources, offering the opportunity for peer review and/or comment) and not be too dependent on complex decision making processes [1].

We might also note that in project-based learning, an open-ended task that supports student creativity in producing the solution is important.

Kear [1] raises some interesting questions: What interactions are there between group members and how do they affect learning? What factors make a successful group project? We explore these further in Section 3.3.

2.2 Assessing teamwork

The method of assessment has been found to be a major influence on the success of group-based projects [7]. Distance learning students can be strongly influenced by the continuous assessment components of the course and use them to guide their own studies. Biggs [8] uses the term ‘backwash’ to describe how continuous assessment biases the course curriculum as perceived by the student, with students concentrating more on those parts of the course that are covered in the continuous assessment material. It should be noted here that, for those unfamiliar with distance education, continuous formative and some summative assessment materials tend to be delivered to students at the same time as their study material and is thus available to them in advance of actually studying the material. By reading the assessment material before studying the instructional course materials, the students can potentially bias the way in which they approach that course material, as they search for answers to the questions at hand, rather than addressing it in a more general or open-minded fashion.

The group projects presented in Technology courses authored within the Department of Telematics have focussed on the group production of an assessed document or report. The reports have taken a variety of forms of presentation in several different courses: for example, as linear (paper-based) report, or as a set of hyperlinked HTML pages viewed in a web browser.

The key issue is the assessment of the group process/dynamic, as well as the end product. All group communications must be held in the group conference area. Although all the group members should be active participants in the group activity for the duration of the group exercise, it is important that some flexibility is maintained (not least because this is one of the major attractions of distance learning) and this means that on occasion one or more group members may be absent from online discussions for an extended period of time. The asynchronous nature of conferencing does offer a flexible approach, but it is possible to get out of synch in a thread, or miss a flurry of activity. Note that if real-time electronic communication, such as using internet chat, is supported, it creates time constraints on participation and such methods of communication are also likely to be missed by the assessor.

The tutor uses the conference transcript to rate the effectiveness of the group process, but does not contribute to it – the students must therefore moderate their own use of the conference, another important skill for the electronic workplace. Students need to know that the tutor knows what is happening in the group (e.g. that someone isn’t pulling their weight, or that someone is doing their utmost to facilitate effective group processes/dynamics) and will take this into account in awarding marks. The availability of the conference transcript is an advantage the marker has over those marking face-to-face group processes [1].

Students are told the breakdown of marks as part of their assessment material and can thus see that getting the group to work will have a positive influence on their individual mark.

Students do not want their marks pulled down by the other group members, but the task must be seen by the students to be a group task (e.g. by virtue of the marks awarded and the structure of the marking scheme).

T209 awards marks for student comments/criticism/feedback, meeting handover deadlines and reacting appropriately to feedback. A summary of the assessment strategies is shown in Table 1.

Table 1 Breakdown of assessment in technology courses with group work component

<i>Course</i>	<i>Individual marks</i>	<i>Group marks</i>
<i>IT and Society</i>	Structure/organisation	30
	Use of evidence/ resources	20
	Contribution to group/discussion	20
	(Total 70)	(Total 30)
<i>Communicating Technology (year1)</i>	70	30
	(Total 70)	(Total 30)
<i>Communicating Technology (year2+)</i>	Individual pages	60
	Account of collaboration process	20
	(Total 80)	(Total 15)
<i>T209 (year 1)</i>	Individual contribution to group process	10
	Individual contribution to content of group web page	30
	Hitting draft to conference deadline	5
	Contribution of comments by agreed date	10
	Using feedback from other group members	5
	Reflection on group dynamics, effectiveness of formal group processes, relation to theory	20
	(Total 80)	(Total 20)

3 Problem-based learning as an example of teamworking

Problem-Based-Learning (PBL) typically describes “an educational strategy in which a curriculum is based upon small group, student-centred learning, focussing on practical problems.” [9]. Rhem [10] describes PBL as encouraging “meaning making over fact collecting” through the presentation of “contextualised problem sets and situations”. The outcome of the problem-based learning approach is “learning that results from working

with problems”. Rhem also presents the ‘official’ description of problem-based learning as “an instructional strategy in which students confront contextualised, ill-structured problems and strive to find meaningful solutions.”

The learning environment provided by group work not only requires students to develop and demonstrate their communication skills, but also an understanding of a particular topic. This is particularly evident when students have to teach other members of their group about the outcomes of their own learning [9]; this also relates to the ‘teachback’ approach of conversation theory [11,12] in which the educator accepts that the student has learned a concept when the student can explain it (teach it) back to the educator in their own words.

There are several issues that must be taken into account when designing a PBL course:

- task/project definition
- group support structures (e.g. in distance learning, how will communication between group members be supported?)
- assessment structure.

3.1 Creating PBL tasks

Scenarios for problem-based learning should be open-ended, authentic and relevant to both the student and the workplace. The problem should also promote self-directed learning (in order to encourage students to take some responsibility for their own learning) as well as helping the students apply and integrate concepts and tools relevant to the topic [10,9,13]. We might add that it is important for the task to motivate students to discover and acquire new concepts, skills and tools so that they learn *how to learn*.

Problems or puzzles are often motivating because they provide a reason for both acquiring and, applying knowledge; and they have the inbuilt payoff of satisfaction at having demonstrably succeeded at accomplishing a task. Jadud [13] points out that students often think they are not interested in a subject, when in reality they are actually uninterested in the method through which the material is presented.

Rhem [10] cites Gijsselaar in defining a *bad* problem-based learning task – such a task includes questions rather than student-based learning activities, similar to working from a text book: there is less motivation for self-study.

However, it is often the case that the educator wishes to guide the students’ learning process and often seeks to impose a particular learning journey according to a fixed syllabus. This is in stark contrast to a discovery-led problem-based learning approach (and to a lesser extent resource-based learning) in which students have significant choice and control over their own learning. Teaching facts according to a fixed syllabus is also easier to assess (the range of correct answers to a fixed form of assessment can be narrowly defined).

OU students are assessed using continuous assessment, usually in the form of assignments marked by their tutor and by an end-of-course assessment (ECA) usually in the form of an invigilated examination. Experience of OU courses with ECA projects suggests that they require more time to mark and have a wider variance of marks (e.g. when a project is double marked) than traditional exam-based assessment.

The course T396: *Artificial Intelligence for Technology*, has a quite narrowly defined ECA project, although there is some scope for student exploration within the confines of the project. This project is individually assessed, but all students (approximately 1200 per year) complete the same project. The project is written anew each year.

Students use electronic conferences dedicated to discussion of the project. There are guidelines in place to prevent students from sharing whole chunks of their work with each other [14]. The follow-on course, TM426: *The IT and Computing Project*, allows students to define their own projects, but this approach requires a significant amount of tutor support, from project statement, planning and development, through completing the project itself and ultimately assessing the final report.

Part of the group task is the problem phrasing/definition/setting/statement. This is a precursor to the actual problem solving process. Considerable creativity can be exercised in this phase; for example using brainstorming. Such a process is potentially more difficult if asynchronous communications is only the available channel.

Jadud [13] also remarks how any problem should be accessible to all concerned – that is, it should be easy for everyone to grasp. The problem should be challenging enough to generate interest and offer a wide range of possible solutions; whilst still allowing weaker students to achieve some degree of success – after all we do not want to put people off learning. (Jadud sums this up nicely in the mantra – “*success early, success often*”).

3.2 Assessing PBL tasks

Sambell [9] further suggests in PBL the problem acts as a stimulus for learning and that both the curriculum and the learning process are defined by the learner in order to explore the problem at hand. Rhem [10] sees a group (of around five individuals) as defining their own ‘learning issues’ in response to the problem and allocating roles accordingly. The tasks investigated by Sambell and McDowell were presented as an underspecified brief for a particular electronic product, such as a radio. This approach was intended to mirror a realistic industrial scenario in which the students had to negotiate a more precise specification through discussion and negotiation with their ‘clients’.

The educator was also responsible for setting bounds on the learners’ discovery process (an approach termed ‘guided discovery’) by providing clear learning objectives. These learning outcomes were designed to keep the students broadly on topic, as well as helping them manage their group processes – allocating tasks, planning and keeping to schedules and so on. Rather than simply being the means to generate an end-of-course result, the assessment process was also used to promote learning (cf. [8] backwash effect). The style of the assessment process was further designed to be in keeping with the teaching method *and* the course aims and objectives.

In particular, students kept detailed documentary evidence of their progress through individual project diaries. These recorded details of the task itself (interfaces between the subsystems developed by individuals, design notes and testing procedures etc.) and the group process (individual roles and responsibilities, schedules etc.). The diaries were inspected during weekly tutorial sessions, so that timely feedback could be provided by the tutor, as well as ensuring that schedules were being set and met. Indeed, meeting scheduled deadlines was an explicit part of the assessment. Regular feedback helped students identify their individual learning needs and hence offered them further control over their own learning [9].

Sambell and McDowell [9] report that the problem-based approach was extremely well received by mature students. In particular, the authenticity of the project, its setting and the mode of assessment were more in keeping with their experiences of the workplace, as opposed to traditional exam-based assessment. The problem-based approach provided students with an opportunity to demonstrate valuable real-world skills in a manner that would be impossible during an examination. Comparatively speaking, exams compared poorly on two further counts – exams are not very satisfying and it is often hard for a student to see why knowledge tested in examinations is important for anything other than passing the exam! Conversely, the project setting provided an element of realism and demonstrated the usefulness and potential application, of more abstract academic concepts. Project work can be more flexible in terms of timing than the artificial deadline presented by an examination (notwithstanding the reality of project schedules and handover dates).

As a rule of thumb, a tutor should not *actively* participate in a group conference for fear of disturbing the group dynamics (although the tutor is expected to keep a track of discussions – ensuring that they remain on-topic and preventing breaches of netiquette). In certain circumstances it may be appropriate for the tutor to intervene. Such an occurrence would be akin to opening up a discussion in a face-to-face environment. In groups where discussion appears to have stalled, or where a plea has been made to the tutor for help, the tutor can post a well-placed open-ended question to get the group moving again.

Where development of group-working skills is one of the stated objectives of the course, it is important that the students believe that the assessment takes the exhibited group processes into account [9,1].

3.3 Robotics as the problem domain

Finding a problem domain that satisfies the constraints of being engaging for students *and* assessable by instructors is a challenge for teaching any form of teamwork. Our experience with young children [15–17], secondary school children [18] and mature adults shows that robotics challenges satisfy the desirable PBL constraints of being open-ended yet supporting early and frequent success. This is in contrast with some traditional computing or engineering assignments where the student may not be able to produce a working product and therefore not participate.

The interdisciplinary nature of robotics projects make them well suited to assessment, since students must take on different roles and thus can be assessed both within their role and in how they contribute to the group. In the next section we explore models of team working based on our studies of successful group projects.

4 Effective teamworking models

We have surveyed a range of student project work in computer science and engineering involving team working, including projects incorporating robotics [19–21].

This experience demonstrates that team working succeeds if:

- *The project is well organised and orchestrated:* the task must be clearly defined and marked by deadlines and deliverables. Guidance should be given to participants on how to approach the project, including both the task and the group interaction. A group must be supervised by a person capable of resolving any problems and there must be mechanisms capable of resolving problems that arise.
- *The group should be of the right size:* too large a group and some will not contribute as they can rely on the work of others; too small a group and there may be a shortage of skills or diversity of input, making it difficult for the group to maintain momentum or even solve the problem.
- *The problem being solved is engaging:* the problem should be both rich and reasonably solvable. It should be sufficiently general as to interest the vast majority of participants, be directly relevant to the other course material and add value to the remainder of the course.
- *There is sufficient reward for individual contribution:* some, or all of the mark for the project should come from the student's contribution to the group. There should be rewards for participation even if the task is only partially completed or not completed correctly.
- *The problem being solved is conducive to a group-working solution:* some problems are better solved by individuals; designers of group-working should not try to force unsuitable problems into the group-working environment.

Our experiences of remote team working have illuminated five basic models that illustrate different degrees of collaboration and can be related to the learning of team working skills. These are summarised in Table 2.

We propose to use a mix of loosely coupled approaches for our robotics curriculum.

Table 2 Some models of remote teamworking

<i>Collaboration model</i>	<i>Description</i>	<i>Students experience</i>
<i>cooperative problem-solving</i>	tightly-coupled, synchronous activity; approximates a face-to-face environment, with all the problems that entails	teamwork throughout the development process
<i>divide-and-conquer</i>	mixes synchronous communication (usually used for planning and resolving integration issues) with asynchronous communication and off-line development	collaboration throughout the development process: planning, negotiation, role identification, discourse
<i>component handover</i>	loosely-coupled asynchronous activity where each student completes a part and hands it on to the next student to work on	negotiation and critical skills; development using others' products
<i>component critiquing</i>	loosely-coupled asynchronous activity where each student reads and openly criticizes the code of another	focuses on critical and discourse skills
<i>individual projects that interact after completion (e.g., in competition)</i>	mixes asynchronous (used to code/build robot) with synchronous (each student competes against another, e.g. football)	project work is independent, but students are able to compare and discuss designs/implementations based on public performances

4.1 Challenge-based learning

We have also been developing educational material using a ‘Challenge-Based Learning’ (CBL) approach [18]. This material is particularly suited to teaching teamwork to children through robotics challenges such as RoboFesta (<http://www.robofesta-uk.org/>). Robotics challenges are increasing in popularity throughout the world. We believe that these competitions have latched on to a number of important concepts that should be considered within any educational context.

- participants have a very definite goal and hence reason for building and programming a particular robot
- the need to improve on a design is encouraged by the desire to improve the robot’s performance with respect to the challenge
- friendly competition between groups of students can foster team building
- everyone’s a winner insofar as their robot can complete a particular challenge to some degree.

Challenge-based learning involves groups of people forming a team to design a robot capable of performing a pre-assigned task. The group is largely self-organising, with roles being decided by the members. In all cases a mentor is available to help resolve disputes, whilst experts from the Open University can be contacted for technical assistance.

The challenges can be quite specific – for instance robot soccer involves constructing a robot to play a game of one-on-one soccer. There is a tight specification, detailing maximum sizes for the robot, the dimensions of the pitch and the type of ball to be used. Conversely, the robot dance competition is open to a far greater degree of freedom; the group have to create a robot capable of dancing to a piece of music, the rest is left to the group’s imagination.

The construction process is relatively lengthy; our school groups have completed most of the work after school or at weekends meaning that construction may take several weeks. During the construction process, the group works synchronously (or near synchronously), with individual members of the group concentrating on their particular tasks. There is a great deal of interaction with ideas being presented, assessed and modified. Perhaps remarkably, most groups work smoothly with little friction.

At the end of the construction process the robots are brought to a central location where they compete against one another. During this part of the challenge, robots cannot be further modified and the success or otherwise of the group is determined by their robot’s results. Each robot has been constructed more or less in isolation to meet a common challenge. Students are encouraged not only to study their own robot, but also to try and determine the strategies used by their opponents.

5 Robotics projects: past and future

The OU course on mechatronics, that ran from 1994-2001, had a Lego-based robot with on-board computer designed and manufactured by the University. Some six hundred of these kits were produced and used annually by a generation of third level undergraduates.

This experience shows that robotics can be taught at a distance and that we achieved the same benefits as reported by others in teaching our undergraduates principles from science, technology, engineering and mathematics.

5.1 Proposed methodology

We propose to try the following model in our third-level course on designing intelligent machines. We shall require students to:

- 1 conduct a group design project
- 2 distribute subtasks in the team
- 3 conference and manage the team project
- 4 build, test and critique subsystems
- 5 assemble the subsystems into the whole robot
- 6 build and test the whole robot
- 7 compete with other teams.

How can we satisfy these objectives? The first three are well-established in the OU system through courses such as TU170: *Learning Online: Computing with Confidence*. The others have been demonstrated through other remote computing projects [20].

Robot design tasks will be formulated so that they can be broken down into subtasks. For example, one student might be responsible for the sensing subsystem, while another might be responsible for designing the gear train and propulsion subsystem. Other students may be responsible for programming and so on. Each student builds a piece of the whole and presents it to the team for *informed* criticism, before repeating the design cycle. This is directly comparable to the tasks performed in traditional universities and in industry.

Every student will be expected to build and assemble all the components using the instructions and program code provided by their fellows. Every student would have a more-or-less identical copy of the group robot that can be used to perform a task. Establishing that the copies *are* comparable will provide practical lessons in benchmarking and testing. Experience leads us to believe that this is a feasible approach; we plan to evaluate it in detail in a series of experiments with remote students.

Finally, how can the teams' robots compete remotely? We expect to set the students a task that can be monitored objectively by their own computer in their own home, with results being assembled over the internet. Again there are exciting possibilities for a web-enabled competition finals day, with teams competing in real-time against other teams.

Assessment is an important question, especially in distance teaching. We agree with Beer *et al.* [2] that the result of the competition is relatively unimportant; it is the keeping of design notebooks, the quality and originality of the designs, the quality of the analysis and the students' reflections on the team-design process. We already have well-established procedures for assessing these factors.

6 Summary

Using the principles developed above we describe a sequence of distance-taught courses using robotics and utilising teamworking across the range of models to teach key engineering concepts.

We aim to construct a course that appeals to a cross-disciplinary population of students. We believe that robots provide an excellent opportunity to interest non-specialists in subjects that they might normally consider too difficult, too technical or too abstract. By their nature, robots are capable of teaching principles of general engineering, computing (including artificial intelligence), technology and even provide insights into animal behaviour and psychology.

References

- 1 Kear, K. and Heap, N. (1999) 'Technology supported group work in distance learning', *Active Learning*, July, Vol. 10, pp.21-26.
- 2 Beer, R.D., Chiel, H.J. and Drushel, R.F. (1999) 'Using autonomous robotics to teach science and engineering', *Communications of the ACM*, Vol. 42, No. 6, pp.85-99.
- 3 Petre, M., Carswell, L., Price, B.A. and Thomas, P. (1998) 'Innovations in large-scale supported distance teaching: transformation for the internet, not just translation', *Journal of Engineering Education*, Vol 87, No 4, pp. 423-432.
- 4 Griffiths, R., Woodman, M. and Robinson, H. (1999) 'Group working for budding software developer', *Paper presented at EdMedia '99*.
- 5 Price, B.A. and Petre, M. (1997) 'Teaching programming through paperless assignments: an empirical evaluation of instructor feedback', *Paper presented at Integrating Technology into Computer Science Education, ITiCSE 97*, June, Uppsala, Sweden, pp. 94-97.
- 6 Price, B.A. and Petre, M. (1997) 'Large-scale interactive teaching via the internet: experience with problem sessions and practical work in university courses', *Paper presented at the ED-MEDIA 97 and ED-TELECOM 97*, June, Calgary, Alberta, Canada.
- 7 Kear, K. (1999) 'Assessment of remote group work', *Paper presented at the What have they learned? Assessment of Student Learning in Higher Education*, 22-23 April, Delft.
- 8 Biggs, J. (1999) *Teaching for Quality Learning at University*, Buckingham: Open University Press.
- 9 Sambell, K. and McDowell, L. (1999) 'Integrating assessment and learning via problem based learning in electronic engineering: student perspectives', *Paper presented at What have they learned? Assessment of Student Learning in Higher Education*, 22-23 April, Delft.
- 10 Rhem, J. (1998) 'Problem based learning', *National Teaching and Learning Forum*, Vol. 8, No. 1, pp.1-4.
- 11 Pask, G. (1975) *Conversation Cognition and Learning*, Amsterdam: Elsevier.
- 12 Laurillard, D. (1993) *Rethinking University Teaching: a Framework for the Effective Use of Educational Technology*, London: Routledge.
- 13 Jadud, M. (2000) 'TeamStorms as a theory of instruction', *Paper presented at the IEEE Conference on System, Man & Cybernetics*, pp. 712-717.
- 14 Weller, M.J. and Hopgood, A.A. (1997) 'Implementing a learning model for a practical subject in distance education', *The European Journal of Engineering Education*, Vol. 22, pp.377-389.
- 15 Johnson, J. (2002) 'Children, robotics and education', *Paper presented at The Seventh International Symposium on Artificial Life and Robotics (AROB 7)*, 16-18, Jan Beppu, Oita, Japan.

B.A. Price, M. Richards, M. Petre, A. Hirst and J. Johnson

- 16 Johnson, J., Hirst, A. and Garner, S. (2002) 'Learning from designing the Robofesta - Blue Peter robots', *Paper presented at The Seventh International Symposium on Artificial Life and Robotics (AROB 7)*, 16-18 Jan, Beppu, Oita, Japan.
- 17 Johnson, J.H. and Hirst, A.J. (2001) *The RoboFesta - Blue Peter Robot Design Competition* (2) Milton Keynes, UK: RoboFesta Edutainment and Robotics Education Research Centre, Faculty of Technology, The Open University.
- 18 Hirst, A.J., Price, B.A., Petre, M., Johnson, J.H. and Richards, M. (in preparation) *Challenge-Based Learning: Robotics Education Through Robofesta Competitions*. Milton Keynes: The Open University.
- 19 Fincher, S., Petre, M. and Clark, M. (2001) *Computer Science Project Work: Principles and Pragmatics*, London: Springer-Verlag.
- 20 Daniels, M., Petre, M., Almstrum, V., Asplund, L., Björkman, C., Erickson, C., Klein, B. and Last, M. (1998) 'RUNESTONE, an international student collaboration project', *Paper presented at the Foundations In Education*, November, Tempe AZ.
- 21 Last, M., Almstrum, V., Erickson, C., Klein, B. and Daniels, M. (2000) 'An international student/faculty collaboration: the RuneStone project', *Paper presented at ITiCSE'2000*, Helsinki.