# COLLABORATIVE OPEN OBJECT LEARNING ENVIRONMENT

**Dr. Colin B. Price**

(United Kingdom, Worcester)

*We have recently developed Java applications which can be deployed on the Internet to provide collaborative learning between students and tutors at remote locations. Students can engage in group activities online and collaborate with tutors in a Distance Learning mode. There are three applications: First a digital circuit simulator which allows students to collaborate in building simple or complex electronic circuits. Each student sees a common shared window on which he may add components to the circuit. There is a built-in chat room to allow collaborative dialogue. Each student also has a private window on which she can build an independent circuit, finally this window may be shared with the group. The second application facilitates collaborative computer programming. Each student programs an individual robot in Java or C, each robot is placed in a shared world which is visible to all students in real time. The third application consists of collaborative building and simulation of non-linear differential equations. We intend these applications to facilitate learning between groups at one or many institutions and indeed provide international collaboration. We also intend to research the establishment and behaviour of collaborative learning groups. This development will involve the addition of Intelligent Agent technology to our applications. We shall make our software freely available to interested researchers.*

## INTRODUCTION

There is a trend in University Education to offer Internet-based education. Research has clearly shown that the Web is an effective learning medium, with student outcomes at least equivalent to those of traditional classroom-based students [8, 9, 13]. As we enter an era where the Web is changing from a medium dedicated to content display to one which is endowed with meaning [3] this new Semantic-

Web will offer new opportunities to researchers in pedagogy. One contemporary and important use of the Web is Open and Distance Learning (ODL). From a pragmatic point of view, there is much to offer here: Students dispersed over a wide geographical area who may be unable to travel to a physical centre of education can register to study in a virtual community. But there is also a clear pedagogical dimension. Learning as a distributed community allows us to refer to the social component of learning as described by Vygotsky [18] as we consider how students can work together in an ODL framework to create new knowledge *collaboratively.* Contemporary extensions of these ideas are found in the notions of Lipman's "communities of inquiry" [12] and Wenger's "communities of practice" [19]. This paper is concerned with developing specific tools to research various modalities of collaborative learning. In section 2 we provide an overview of significant baseline research into ODL and especially collaborative learning and identify one particular shortcoming of current developments which this paper will address. In section 3 we outline the structure of the software we have developed to support our proposed extension to collaborative approaches and in sections 4 to 6 will detail three tools we have developed to research our paradigm. Section 7 returns to the pedagogical issues relevant to this study and Section 8 provides a round-up, critical analysis of our paradigm and suggests where developments need to be made.

**APPROACHES TO OPEN AND DISTANCE LEARNING**

Current thinking in education maintains strong interest the *Constructivist* paradigm of learning. Discussions on collaborative learning should start from this base [10, 14]. Here, collaboration gives learners real-life experience of working in a group, learning from others and contributing their own understanding. Students perceive the need to be responsible for the veracity of their contributions which leads to improved personal learning. Collaborative learning may be defined as "… a situation in which people learn or attempt to learn something together" [5]. This has always been the case throughout history, first through personal contact, then through written correspondence, through telephone, and now via email and the Web. The challenge to the Computer Science community is to research and develop tools to facilitate electronic collaboration.

There is a plethora of tools currently available to support electronic collaboration and a number of reports on these tools have appeared in the literature. One example is the Athabasca University Centre for Distance Education report which presents the results of an analysis if over 100 collaborative tools [1]. Trends identified in this analysis are the use of text-conferencing, audio-conferencing, video conferencing, the use of whiteboards, polling tools and entire course delivery tools. Within Europe the "Innovative Technology for Collaborative Learning and Technology Building" (ITCOLE) project was aimed to develop pedagogical models of collaborative knowledge building for European education. It also aimed to produce a modular knowledge-building environment to support collaborative learning [11]. The focus was clearly on the pedagogy, not the technology. Aimed at both primary and secondary schools, this initiative was designed to develop students' abilities to adopt, cultivate, create new and share knowledge with others. The guiding premise is that knowledge is not absolute nor static rather it is shared within social organizations, and that learning occurs within such communities. It proposed a move away from existing e-Learning environments which were designed to manage study materials, the students themselves, assessment, grading and basic cooperation, towards encouraging engagement in active learning and knowledge creation. Interestingly as we expected, the participating countries all adopted the same constructivist approach to education yet their various historical approaches to education could not be discarded. The ITCOLE project has developed a number of significant tools. "Synergeia" is an extension of BSCW which needs no further comment, "Fle3" is a VLE which provides the usual access to shared resources, a structured "Knowledge Building" tool which is an interactive prompting database, and a "Jamming" tool which is a shared space for the construction of digital artefacts (video, pictures, text, etc.) [11]. The collection of tools is highly structured and contains many management facilities.

An alternative inroad into discussing the collaborative ODL is a consideration of *science education* as a special case. Science education may be viewed as an important target for ODL since the natural process of scientific enquiry involves collaboration (and competition!), peer review as well as the activities of experimentation and simulation which complement the activities of theoreticians. A huge number of demonstration and simulation applets have appeared in

recent years together with toolkits for building these applets. Pupils in secondary education (eg US K-12) whose teaching was traditionally based on acquisition of a corpus of scientific facts are now, via these simulations, able to participate in the scientific method, engaging in the interplay between theory, experimentation and simulation. The "CoVis" project at NorthWestern University is a good example in science education where there are a range of collaborative tools including video teleconferencing, a shared software environment which includes visualisation tools [4]. Significantly, teachers are involved in the production of these tools.

We propose that the domain of science education is rich and potentially fruitful to research collaborative ODL. This follows directly from the *scientific method* used by practising scientists and promoted my contemporary science instruction. Here the thrust is an understanding that a priori models can be tested and falsified in the sense of Popper which in turn leads to the development, extension and integration of a theoretical corpus of knowledge. Objectivity is ensured by the process of peer-group review and publication. These meta-concepts (or rather the development of students' awareness of them) may well be related to the development of autonomous and collaborative learners. Collaboration exposes the individual learner to alternative solutions and approaches to solving problems. Collaborative experimentation and discussion implies a real-time peer review process. Students cannot side-step these issues, rather they become intimately involved in these meta-concepts which appear as real concerns as they are collaboratively solving a problem in a concrete domain such as digital electronic circuits, programming or linear systems.

Turning to the techniques involved in collaborative learning (and the software developed to support them) we find the following classification: Discussion Groups (email BSCW), systems for Data Collection & Organization, Sharing Documents (eg "SamePage" [6]), Synchronous Communication (such as online chat and video conferencing) as well as large-grain Online Courses or Workshops. We propose to classify these approaches as providing a "broad" form rather than "deep" form of collaboration. They all include some management functionality addressing the issues of access into a collaborative asynchronous or synchronous group. Research and development of these systems started from *system* perspective, rather than the *content* perspective. As educators dealing with subject content,

we propose to approach the development of collaborative ODL materials from the fine-grain of the subject content, at least where this is possible. This is our definition of "deep" collaboration which will become clearer in the sections below. Typically we have engineered a number of distributed simulations, where students form collaborative groups to synchronously work on one or more specific problems. Typically the software application provides them with a modelling window, where they may drag and drop graphical objects, write computer code, or compose differential equations, and a second visualization window, where they may view the results of their simulations. We have designed various modalities on this theme: Students may autonomously build a model then share it with their peers who are able to simulate it and review it, students may share the model with the group in real time as the model is built interactively *by the group*, and finally students may autonomously build a model which is published in a shared group environment with all other group member's models, whereupon the collection is simulated together. The latter modality is used in our collaborative (or competitive!) robot programming exercises. This work is the result of a number of separate initiatives [15,16]. All the facets of collaborative ODL mentioned above are apparent in this "deep" paradigm. But here the students' learning is not deflected by ODL management issues, the ODL is hidden technology not visible to distract the student. Collaboration emerges naturally from the focus of learning around the subject content. We propose this is to be most natural, fruitful and fun locus for collaborative learning.

**SOFTWARE STRUCTURE OUTLINE**

Java is used to provide a lightweight accessible Client-Server architecture where each student's Java application is configured as a client. Since our current (and future) areas of application are all *dynamic* (ie involving a simulation), the locus of the simulation engine is a major concern. In order to distribute the required processing, this engine is placed within each client. The actual objects of simulation may be quite complex and involve graphical information, eg circuit elements for the Digital Simulator, and Robots and other RobotWorld objects for the Programming Educator. It is highly desirable to transfer the "minimum number of bytes" between client and cli-

ents via the server. Currently we employ the Java Serialization interface, using Remote Method Invocation (RMI) as a generic communication methodology. The software is designed so that each simulated object communicates a minimal serializable "footprint" to the server. The server maintains lists of these "footprints" and distributes them to the clients when appropriate (see Fig.1). Typically a footprint will include the state of the object, its location on the clients' visualisation screen and a timestamp. It is the responsibility of the clients to reassemble the object from the received footprint and to execute the simulation. This software structure ensures a most efficient distribution of processing and a minimal requirement on Internet bandwidth.

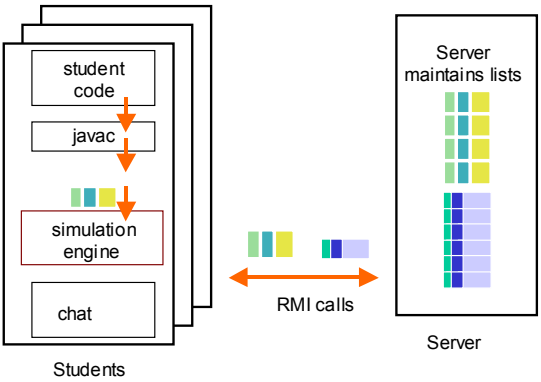


**Figure 1**. Student machines share information stored on lists in the server. Chat lists and application lists are separate. Each student has a simulation engine and a javac compile (when needed) on his local machine.

**THE DIGITAL SIMULATOR**

The Digital Circuit simulator is aimed at teaching and learning in first year (CS1) Computer Science courses as well as in Secondary (K-12) schools. It provides the ability to construct and simulate both combinatorial and sequential digital circuits of high complexity. Circuit elements include the standard logic gates, counters, shift registers, memory and ALU blocks. The click and drop interface allows rapid construction of a digital circuit, a number of input sources

(switches, oscillators) and output devices (LEDs, oscilloscope) allow meaningful circuits to be assembled and simulated. Naturally the simulation is not at the detailed level seen in products suited to Electronic Engineering courses (such as pSpice) and does not model components at the individual transistor level. Circuit element delays are not incorporated, rather the focus is on combination of the standard logical blocks. The user interface is seen in Fig.2 Where the student clicks and drops circuit components onto a "workbench" canvas and connects these with a choice of connectors. The constructed circuit is immediately available to simulation by the student. Students may share their circuit within the learning group by transmitting it to the system server which in turn distributes the circuit to all group members. Each transmitted circuit appears as a separate workbench in a Java tabbed pane. Students receive each others circuits and are able to directly simulate these, make modifications and retransmit them to the server. Coordination of this group activity is achieved through an integrated chat window which allows individual students to inform the group (or post requests to the group) concerning the current activity. In a typical work session, each student will see his own circuit and, via the tabbed workbenches, the circuits built by the other group members. These circuits may result from a task set by a tutor or have been developed by students themselves as they respond to a problem set by the tutor, e.g. "develop a controller for a slot machine". Digital electronics like most engineering disciplines supports multiple solutions to a particular problem specification. It is in such a context that collaborative learning can honestly emerge, as students perform critical analysis of various solutions proposed by group members. Note that these solutions appear "quasi-synchronously" in that all members of the group are together "online" and work to solve a problem. This modality emphasises the peer-review character of "science". But it is also possible, once the group has gained confidence as a group, to move into an "engineering" paradigm of teamwork. Here we envisage the group to divide a more complex problem into components, and each member (or subgroup) of the group to attack and solve a particular sub-problem. A typical complex problem could be the design of a traffic control and monitoring system.

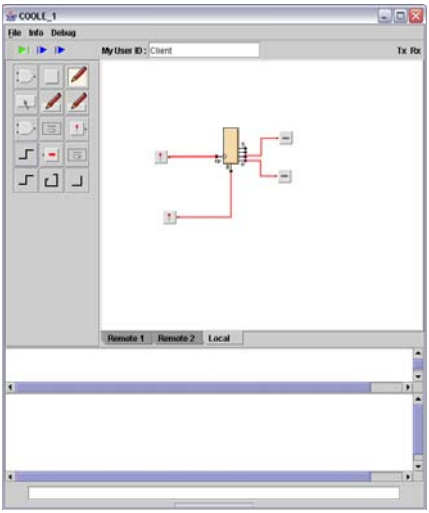


**Figure 2**. Digital Electronics Workbench. Here the local student has constructed the digital circuit. He transmits this to all students in the group who are able to simulate the circuit on their local machine. The windows at the bottom provide the chat-room facility.

The pedagogical implications of this modality of collaboration are clear to us. At present we employ a non-Internet modality of collaborative learning: Following a lead lecture, students are dispersed in groups to solve particular problems in Digital Electronics in a workshop activity. This is followed by a "plenary" session where each group of students may elect to present the results of their activities to the entire session corpus. Here, critical analysis of the groups' contributions is made, and the guiding tutor attempts a convergence of ideas to a "reasonable" (or "optimal") solution. There is limited possibility to combine efforts from the individual groups. This presupposes that all students are available for some three to four hours in the same physical location. Our collaborative ODL paradigm removes this restriction: Students may collaborate over the Internet internationally, and may arrive at the same shared conclusions as our physically located groups.

**LEARNING PROGRAMMING VIA COLLABORATION**

Over the past year, the Computing Section at UCW has developed an Integrated Learning Environment for learning programming (both Java and 'C') using autonomous robots as the vehicle for learning. The paradigm of this approach has been detailed elsewhere [17]. We have recently extended this approach to take advantage of pedagogy implied by the collaborative ODL paradigm [16]. The scenario is a "Robot World" where various active objects such as robots, rocks, trees and gremlins move around a shared space. Each student is asked to program the behaviour of his robot within this world to achieve a particular goal, such as seeking a light, or clearing up some rocks into piles. Each student develops some Java code which is compiled into the controller of his own robot. Our system then places each individual student's robot into a common world and students can observe and evaluate the performance of their particular programmed controller in the World which includes robots from all students in the group. The learning group observe a "deep" simulation in a World where each robot is present, and may modify their programmed code according to their evaluation of the performance of their work. The collaboration is "deep" in the sense that each student observes a common world which contains the results of each student's work. Modifications to behavioural code occur in "real-time", supported by chatroom interaction, hence learning occurs at a "real-time" pace which is impossible to support by conference technologies.

**COLLABORATIVE MODELLING USING DIFFERENTIAL EQUATIONS**

The modelling of dynamic systems using ordinary differential equations (ODEs), despite its long history in science (over 200 years) remains an important paradigm. Alternative methodologies such as Discrete Event Simulation (DEVS), Finite State Machine, Markovian processes and Petri Nets, for example, have become important in modelling Natural systems, Business Information Systems and Computer Science amongst others, nevertheless ODE models remain a fundamental tool for modelling natural systems including population dynamics, economics, and biophysics. Experience gained and confidence in the use of the mathematics of ODEs is still valid and viable. To support collaborative ODL work using ODEs we have developed

a Java tool to support the real-time interactive collaborative building of ODE models for dynamic systems. Realistic models such as those describing the behaviour of single or aggregates of biological cells, neural circuits or economies can take the form of a variety of systems of equations. Each system is open to the specification of a number of parameters, which may be inferred from expert knowledge of the domain, via hypothesis based on theory, or via experimental results. Clearly there is a need for simulation here, to critically analyse the results of proposed models against experimental results. This perspective has provided motivation for development of a collaborative ODL methodology specifically aimed at the development and analysis of systems of ODEs. Take the case of a model of a biochemical system. The tutor may provide the student with a system of ODEs corresponding to an enzyme reaction. He will be required to perform a parameter search of the system looking for fixed-point or oscillatory solutions. Our collaborative software allows students to work with a common set of ODEs and to explore the parameter space collaboratively, sharing their discoveries and questions via the chat room interface.

In a higher-level task, the tutor may ask the student group to model a particular system, e.g. problem in non-linear oscillation. Here the students will not be provided with the ODEs but must generate them from the relevant physics involved. Each student within the collaborative group will construct a set of ODEs and simulate this set and of course compare its behaviour with the target experimental data. When satisfied with his results, these will be published to the group, to elicit comment and feedback via the chat facility. Each student receives the submitted equations (and parameters) and is able to perform an analysis of the received system. In effect, each student is able to perform a dynamic "peer review" of his fellow students; hypotheses. The parallels with the operation of the *scientific method* are clear.

The student environment is sown in Fig.3. In the left window the ODE is entered in a simple and transparent Java syntax and on the right the solution of the ODE is obtained. A variety of numerical solvers (eg RKF5) are provided. On the right various projections of the phase plane are provided visualizing the results of the simulation.

The ubiquitous chat interface is present at the bottom. Typically, each student (or pair) will work autonomously investigating aspects of the model provided by the tutor. This may involve a parameter study of a given set of ODEs or else the construction of a set of ODEs to model a given system. When confident, students will submit their solution to the group, using the chat facility as a mediator. Then critical analysis and peer review will occur and students will take on board the knowledge gained from this process to update their own model. Clearly the process is iterative, and unlike teaching directed to one common goal or "correct answer", this modality encourages students to produce distinct models and solution to a particular given proble. Higher level learning is facilitated by this approach.

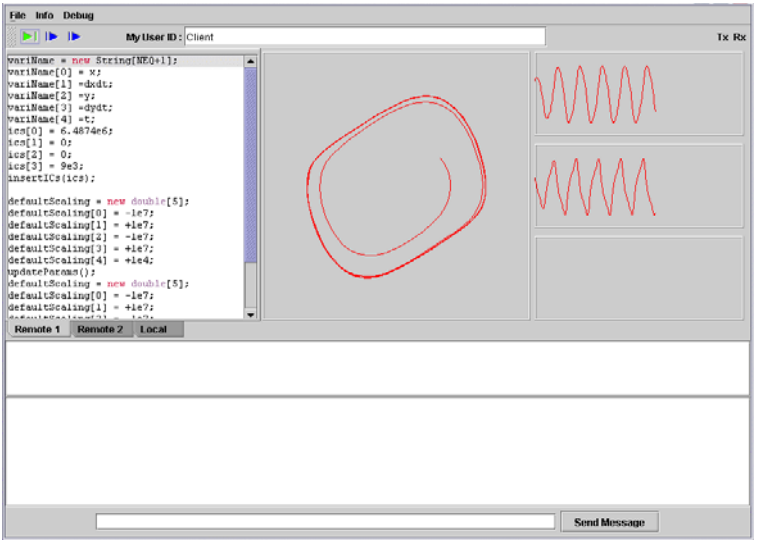


Fig.3 Collaboratiuve Environment for solving IDEs. Here a student has simulated a Van der Pol oscillator from the code (left window) posted by a second remote collaborating student.

**PEDAGOGICAL ISSUES**

We have approached the issue of collaborative group work from the scientific and engineering perspective where we have argued that

the nature of work within these disciplines is collaborative. There are other cases where collaborative *teamwork* is the norm, e.g. with the health care disciplines. Here it is important that health care students learn to work with people from different backgrounds and also with international students. The Occupational Therapy Internet School (OTIS) funded by the EU aims to promote international collaboration within health care education. Using agent technology they attempt to form synchronous learning groups [2]. We expect that in today's climate of globalisation and convergence, collaborative ODL will emerge as the norm in many areas of education. So one must finally turn to the pedagogical, and embrace the theory of collaboration. A good starting point is with Dillenbourg who identifies various factors in education which bear directly upon design of collaboration models [5]. These are the *constructivist* approach which emphasizes the individual learner, the *socio-cultural* approach which emphasizes the relationships between the individual learners and the *shared cognition* approach which emphasizes the importance of the learning materials and learning context. How does our *content-based* notion square with this widely accepted theory? Clearly very well. Except perhaps for the socio-cultural dimension (which is based on approaches of Piaget and Vygotsky), where as yet at UCW we have no experience of international electronic group collaboration.

**CONCLUSIONS AND FUTURE DEVELOPMENTS**

We have described the results of a software development project which has produced three applications providing collaborative ODL tools. We intend to rapidly deploy these tools in short-term trials and use the feedback to drive the next phase in the development of our collaborative-ODL arsenal. This phase, which has already commenced, involves the use of agent technology to assist in the automatic formation and maintenance of international and national collaborative groups. We expect to deploy once again with Java technology, using the Jade agent platform [7]. We especially look forward to working in close collaboration (!) with our Russian partners, especially at Moscow State University.

**REFERENCES**

1. Athabasca University Centre for Distance Education.2004 URL http://cde.athabascau.ca/softeval/reports.htm
2. Beer M.D. and Whatley J. "A Multi-Agent Architecture to Support Synchronous Collaborative Learning in an international Environment. Proceedings of the first international joint conference on Autonomous agents and multiagent systems: Bologna, Italy 2002
3. Berners-Lee, T. (1999). Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor. San Francisco: Harper
4. "Covis Learning through Collaborative Visualization" URL http://www.covis.nwu.edu/
5. Dillenbourg,P. "What do you mean by 'collaborative learning'", Collaborative Learning: Cognitive and Computational Approaches pp1-19, Oxford:Elsevier 1999
6. Electronic Collaboration: A Practical Guide for Educators 1999. The LAB at Brown University.
7. Jade URL http://jade.tilab.com.
8. Gerhing, G. (1994). "A degree program offered entirely online: Does it work?" In D. Foster & D. Jolly (Eds.), Proceedings of the Third International Symposium on Telecommunication in Education, (pp. 104-106), November 10-13, 1994, Albuquerque, New Mexico.
9. Golberg, M. (1997). "CALOS: First results from an experiment in computer-aided learning for operating systems". Proceedings of the ACM's 28th SIGSCE Technical Symposium on Computer Science Education (pp. 48-52), San Jose, California.
10. Hooper, S., & Hannafin, M. J. (1991). "The effects of group composition on achievement, interaction, and learning efficiency during computer-based cooperative instruction" Educational Technology Research and Development, 39(3), 27-40
11. ITCOLE Report 2000 URL http://www.euro-cscl.org/site/itcole/ITCOLE_Final_Report.pdf
12. Lipman, M. (1991). Thinking in education. Cambridge: Cambridge University Press.

13. McCollum, K. (1997). "A professor divides his class in two to test value of online instruction." Chronicle of Higher Education, 43(24), A23
14. Palloff, R. M., & Pratt, K. (1999). Building learning communities in cyberspace. San Francisco: Jossey-Bass
15. Price, C.B. Breakfast with Dr.C – Systems of First and Second Order Differential Equations Acco, Leuven 1997
16. Price, C.B. "An Integrated Programming Environment suitable for Distance Learning" Proceedings Frontiers In Education Conference, Savannah, Georgia 2004
17. Price, C.B "An Integrated Programming Environment for Undergraduate Computing Courses" Proceedings **International Conference on Information Technology: Research and Education** London Metropolitan University October 2004
18. Vygotsky, L. S. (1978). Mind in society, the development of higher psychological processes. Cambridge, MA: Harvard University Press.
19. Wenger, E. (2001). Supporting communities of practice: A survey of community-orientated technologies (1.3 ed.) (Shareware). URL http://www.ewenger.com/tech