From overhead projection to effective interactive learning software for science students

Ralf Cord-Ruwisch
Murdoch University
cord@central.murdoch.edu.au

Ralf Cord-Ruwish is program chair of Biotechnology at Murdoch University's School of Biological and Environmental Sciences. His research interests include the physiology of anaerobic microorganisms, bacterial corrosion, and microbial physiology.

Summary

Small, simple, highly interactive software modules have been developed to help science students surmount some of the main barriers for effective learning (as experienced in traditional lecture-based teaching): lack of motivation, lack of rapid and specific feedback, lack of interaction with the learning material, and lack of the opportunity to apply newly acquired knowledge.

These modules are different to other multimedia materials currently available in that:

- they will allow students to progressively test their assumptions and further their understanding of scientific concepts. Students will develop not only conceptual understanding but also problem-solving skills;
- the use of academics as software developers means that the modules can be easily changed in response to student needs and feedback;
- students' progress and difficulties can be tracked and students will be able to record comments while using the modules. Tracking and students' comments will allow teachers to discover areas of difficulty which can be addressed through small group work and also provide for a continuous cycle of development, use and evaluation;
- the modular construction makes sections of the program interchangeable between disciplines and allows academic staff to build courses by drawing on small modules of relevant content instead of large blocks of content comprising significant amounts of irrelevant material.

Current learning situation

In many universities, science teaching is based on traditional lecturing with end of semester examinations. Usually this results in students being preoccupied with note taking in lectures rather than reflecting on the content. Proper interaction with the material and intensive studying often does not occur until the examination period. Typically students study a large quantity of material at the end of semester resulting in shallow learning with low retention, poor problem solving skills, and unsatisfactory conceptual understanding.

Changes initiated with the use of interactive software modules

Use of software modules

Science courses at Murdoch University have also generally followed this traditional pattern, but in 1995 an attempt to change this was begun in third year Biotechnology with the transformation of about 10 lectures into highly interactive software modules. In the initial phases of this pilot project students downloaded the modules onto floppy disks and worked through the material as a supplement to the lectures, either at home (80%) or in the university's computer laboratory (20%). Student feedback indicated that they found the material more exciting than lecture presentations and more effective for their learning as they had "time to think about the material and understand it". Following this positive feedback it was decided to replace three lectures completely with the interactive software modules. Built-in progress tracking indicated that 48 out of 50 students worked through these lecture-replacement modules, successfully completing the randomly-generated problems which had been designed to help them progressively develop and apply the concepts on which the lecture would have focussed. Again, students reported perceptions of having learnt more effectively than they did as a result of lectures.

Expected outcomes of including teacher-designed highly interactive software modules in science teaching

Typically, in lectures students are passive, they are not provided with the opportunity to apply concepts as they are being developed or receive feedback on these, and are not well catered for as individual learners. By contrast, as indicated through student response to the modules produced thus far, it is expected that the use of the interactive software modules described will achieve the following learning outcomes:

- *Immediate feedback leading to enhanced learning*: Each module is capable of providing students with immediate feedback in much the same manner as that provided by one-on-one tutoring. Although feedback itself is not sufficient for the development of understanding, feedback that relates to the testing out of assumptions or application of knowledge to a problem is critical to deep learning and is the objective of many good laboratory classes.
- *Problem-solving skills*: The software modules specifically require students to adopt a problem-solving approach when dealing with scientific concepts. In line with adult learning principles students will have to construct their own understandings by referring to both their existing knowledge and the new information presented at each stage of a module. Individual differences in understanding and progress can be catered for because students will be able to work through the modules at their own pace and choose their own learning pathway.
- *Increased motivation and retention*: The modules actively engage students in the exploration of scientific concepts in a way that student involved in the pilot project have described as stimulating and motivating. In progressing through each module they test and become aware of their developing conceptual knowledge and feel rewarded through the visible indication of their progress towards understanding. Motivation is a key factor in learning and retention.

Other possible benefits of teacher designed software modules

- It will reduce the time that teachers need to spent on assessment tasks because of the automated tracking of students' advancements in the modules. This time can be used more effectively in additional student/teacher interactions (*e.g.,* small group teaching). The use of problem solving program modules has been found to result in students being eager to obtain information needed to complete the modules.
- Program can effectively prepare all students for laboratory classes (if attending the lab is subject to having reached a certain level of skill or understanding in the program).
- Fast upgrading, trouble shooting, debugging and accommodating students input. As the teacher is also the developer, the program can be modified instantly upon students' requirements. This is more difficult with traditional multimedia projects (CD-ROM, developer team , *etc.*).

- The authoring of program modules engages the teacher in educational theory and stimulates the teacher to think of students possible misconceptions (*e.g.,* in the design of feedback to incorrect answers) which can be more intellectually rewarding than repeating lectures year after year.

Steps from overhead projections to developing interactive teaching software

Use of computer presentation software

To prevent students from being preoccupied with copying notes, computer generated presentations (*e.g.,* PowerPoint) were used in lectures and copies of all slides provided in the study guide. While this helped students to focus on the content of the lecture, it did not cater for a number of additional students needs such as: individual pacing, building up simple slides to complex graphical diagrams and minimising interference (distraction).

Use of self-paced presentations

It was decided to convert some of the conceptually more challenging presentations onto a platform that allowed the students to view the material at their own pace, either at home or at university computer facilities. *Authorware Professional* was chosen as one of the most user friendly authoring tools for "non programmers". The first attempts with this software resulted in linear paging programs (electronic book) that were clearly analog to the lecture presentations. Students could move from screen to screen by clicking a "Continue" button. This clearly offered students a self-paced approach and the possibility to view the material at the time of their choice. However, after preliminary tests with students, this material was found to be unsuccessful in enhancing students' learning as many students viewed the material relatively superficially, and "skipped from screen to screen in the hope to come across something exciting". It was perceived that this way of passively viewing information had no benefit over the use of lectures or books as the medium. The inclusion of branched pathways and "jumping" to other parts of the program (hypertext links) did not appear to address this problem.

Omitting continue button

In order to force students to take full notice of what is presented on each screen the standard "Continue" button was omitted and replaced by a hidden "Hot spot" on the screen. In the development of graphical representations of biochemical chain reactions the explanatory text on the screen (equivalent to the lecturer's comments) asks students (by using coloured text) to point to a particular part of the diagram. Correct pointing (hitting the hidden hot spot) results in the further development of the diagram while incorrect pointing can trigger a specific explanatory feedback. This relatively small change improved the effect of the program as it allowed only those students that read the comment, reflected about its meaning, and could relate it to the diagram to advance at a reasonable pace. The completion of sentences, the calculation of simple numerical problems and the positioning of items to specific locations were used in much the same way. In comparison with lectures, students can only advance further in this presentation, when they "follow" the train of thought of the author. This could be particularly useful as a replacement of pre-laboratory talks.

Students assisting in the development of the "presentation"

While the approach described above asks the students to follow and to act during the presentation of material, it did not necessarily require them to reflect in more depth and detail on the material. In more recent program modules it was attempted to use the computer program analog to a one-on-one tutoring situation.

After providing students with a little information pertaining to the target concept, the program requires them to test their assumptions about the next logical step by suggesting what this step would be by either keying in a word or numerical answer, plotting a point on a graph, moving an object to a correct position, making a sequence of choices or adjusting parameters in a process simulation. Following a student's response, the student is given different feedback by the program, depending upon whether the response was appropriate or not. If students choose an answer within the range of possible correct answers, they are presented with a new problem that tests their assumptions and challenges them to work out the next logical step in the argument. If the answer chosen is incorrect they are provided with further information that hints at other aspects they need to consider in order to solve the problem or directs them to additional resources within (different module or level) or outside (tutor, textbook, *etc.*) the program. Students learn from the instant feedback to their actions similar to a laboratory situation. It is hoped (but not evaluated yet) that this way of helping to construct knowledge gives the student the feeling of owning some of the knowledge rather than "being filled with knowledge". One of the aims behind using this approach is to include a constructivist approach of student centred learning. Future programs are planned to include further aspects of the constructivist theory (*e.g.,* students decide on which modules they use to obtain tools needed to solve particular problems).

Advancing by problem solving

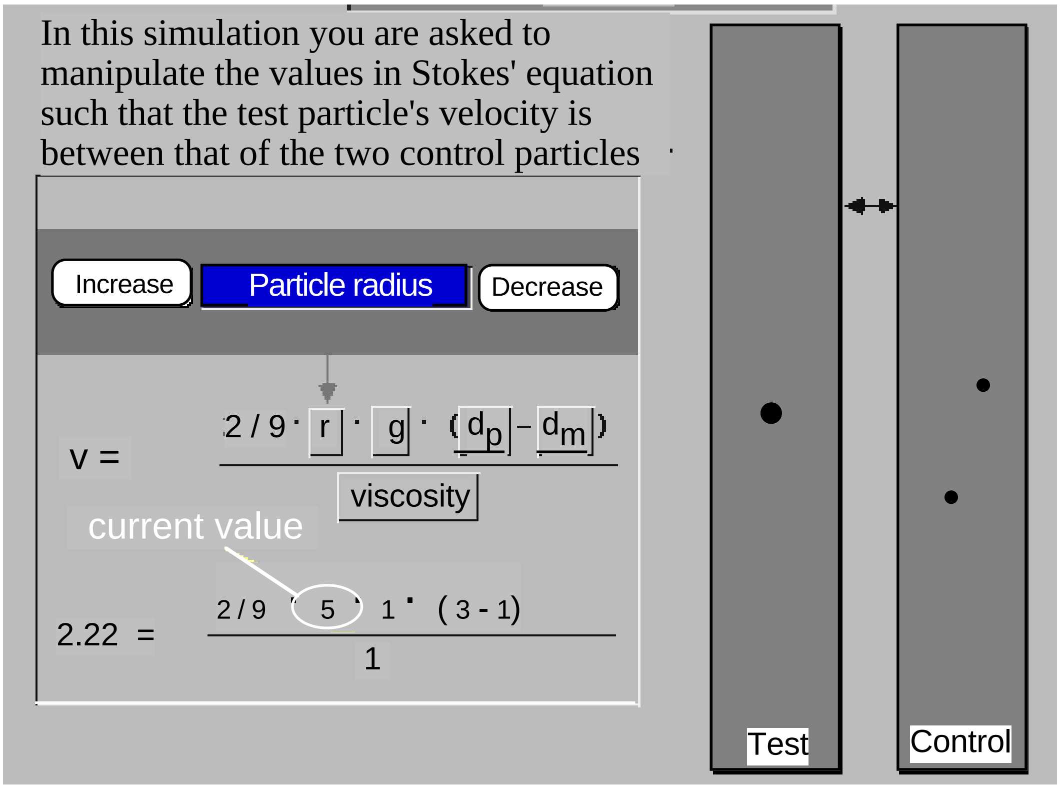
A simple (in terms of authoring skills needed) but very effective way (judged from students feedback) to develop particular skills in students turned out to be the use quizzes. For example, students learn about the oxidation states of carbon in organic compounds. The students are asked for example "What is the oxidation state of carbon in methanol?". A correct answer results in the increase of a visible numerical counter and in a the display of a new compound structure to be analysed. If the wrong answer is given, the student is asked to "Please consider the oxidation state of oxygen (-2) and hydrogen (+1) and the fact that the total oxidation state of this uncharged molecule must be 0". Again the main learning effect is by learning from feedback to mistakes, quite similar to laboratory experiments. It was observed that students are more interested in pieces of information when they first have experienced the need for it. After having achieved say 10 correct answers students obtain a "token" (bonus point) that certifies mastering of a certain set of skills. After developing and fostering different skills in this way, students obtain the tools to solve more comprehensive problems (for example establishing the mass balance of bacterial fermentation processes).

Including a Game Component

The use of tokens or similar rewards was demanded by students after working through a program module without a visible reward at the end. However, it was observed that some students still preferred to guess the correct answers and were not fully focused on "getting it right". By contrast the same persons could be seen very concentrated, focused and aroused when playing commercial computer games. It was attempted to somehow draw on some of the obvious energy that computer games manage to mobilise from within the user. Realising that the obvious fascination with many computer games is not exclusively to win bonus points but also the exposure to the stress of failing (*e.g.,* death, or dropping to a lower level as a punishment), a stress component was included in some parts of the program by requesting students to get 10 correct consecutive answers before a "token" is given. Mistakes result in the loss of all credit points (counter is set to zero). This inclusion of this educationally doubtful "negative conditioning" resulted in markedly higher levels of tension and arousal when students had reached 8 or 9 correct answers and did not want to lose these points by a careless answer. In computer labs they reflected more carefully about the answer and also consulted each other before keying in their answer. Compared to the situation in many lectures, this small change seemed to have students' internal energies focused on the learning objective.

Experiencing scientific laws and processes - relation to dry labs

With some experience (*e.g.,* three months) in developing simple program modules as described above, more complex modules could be developed that make use of some mathematical principles behind scientific laws and processes. For example in a module about the sedimentation velocity of particles, students manipulate the parameters of Stokes' Law and observe the effect of the manipulation on the outcome of a simulated process (*e.g.*, sinking velocity of a particle, Figure 1).



In this simulation you are asked to manipulate the values in Stokes' equation such that the test particle's velocity is between that of the two control particles

Figure 1. Part of the screen allowing students to interact with Stokes law (modifying variables in the law to adjust the sedimentation velocity of a particle to a certain level).

It was perceived that students should not only be given room to freely explore this principle but also be required to solve problems of significance in their subject area. Here, for example, the velocity of a particle had to be changed such that the movement of the particle on the screen fits certain criteria; for example:

the particle velocity reaching a certain value;

determination of viscosity from monitoring the settling velocity in different media;

adjustment of buoyancy and velocity of a planktonic alga by changing the algal floc size, lipid content *etc.*).

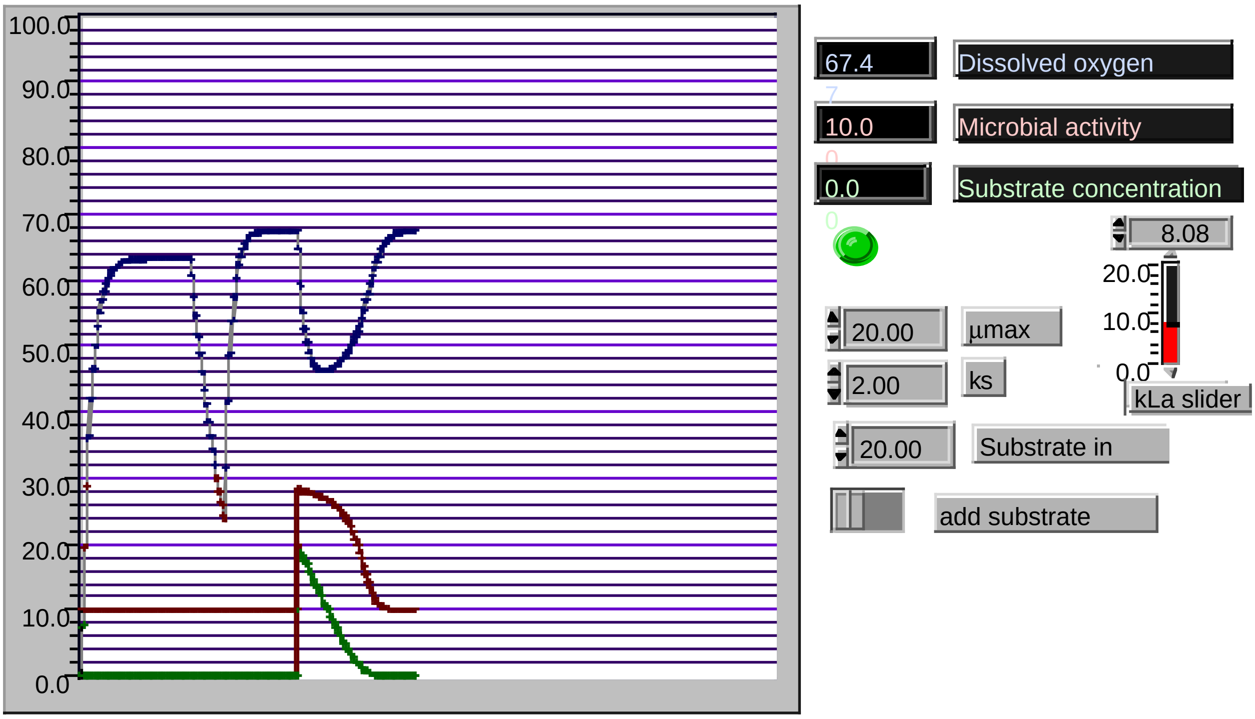
This approach encourages the understanding of the law by applying it. This is similar to the intent of typical laboratory classes. Although it is recognised that such simulations cannot replace the real world experience of experiments they offer a number of benefits in addition to lectures, tutorials and laboratory classes:

- active learning at their own pace;
- learning from feedback;
- effective tool to make sure students have achieved a certain level of experience or skill prior to laboratory classes (no more unprepared students?);

- possibility to allow students to interact with simulated processes that would be too time consuming, expensive or dangerous to run in the lab.

Student real time interaction with simulated industrial or laboratory processes

Other simulations developed engage the students in real time simulation of processes that are too difficult or expensive to run in laboratory classes. An example is the simulation of microbial oxygen uptake kinetics in a chemostat culture as a function of different variables such as the oxygen mass transfer coefficient of the aeration equipment, the air flow rate, the substrate concentration, the microbial kinetic parameters (k_M value, v_{max} value, endogenous respiration rate, oxygen half saturation constant , Figure 2).



67.4 Dissolved oxygen

10.0 Microbial activity

0.0 Substrate concentration

8.08 kLa slider

20.00 µmax

2.00 ks

20.00 Substrate in

Figure 2. Part of the Screen (here shown in black and white only) of students process simulation (addition of substrate to a starving microbial culture, monitoring of oxygen and substrate concentration, and microbial activity).

For some of those simulations that display the current state of a process via a running graph other, faster authoring tools such as LabView had to be used†. A comprehensive simulation of such a bioprocess is given to students for investigation, analysis and evaluation. Students produce an assignment (each student works with a simulation that uses different sets of parameters) that can be easily marked by comparing with values in a programmed computer spreadsheet.

The simulation of a sophisticated bioprocess was used in combination with laboratory project work where students were exposed to the "nuts and bolts" of the bioprocess such as calibrating flow pumps, adjusting oxygen flow, aseptic sampling *etc.* As the simulation always guarantees a set of useful data for analysis and interpretation there is no necessity for the laboratory component to produce "good data" hence students can be allowed to work somewhat more freely and learn from their own mistakes.

† Example modules of such processes can be provided to interested teachers.