Case Study 6

Simulations for visualisation of complex processes and principles in chemical engineering and in physics

Cheryl Hodgkinson-Williams

February 2009

Title: Case Study 6: Simulations for visualisation of complex processes and principles in chemical engineering and in physics

February 2009

Report of the OpeningScholarship Project funded by the Shuttleworth Foundation

Author: Cheryl Hodgkinson-Williams

Cheryl.Hodgkinson-williams@uct.ac.za

Published by the Centre of Educational Technology, University of Cape Town, Private Bag X3, Rondebosch, Cape Town, 7701.

Contents

List of figures 5
Acronyms and abbreviations 6
Introduction to the teaching and learning case studies .... **Error! Bookmark not defined.**
Background: The OpeningScholarship Project .... **Error! Bookmark not defined.**
Executive summary 7
OER opportunities 7
Introduction 7
Open source software for teaching and learning 8
Use of open source software in chemical engineering education 8
Use of open source software in physics education 9
Background and methodology 9
Selection of cases 9
Case study methodology 10
Case study: OSS by Chemical Engineering 11
Contextual and pedagogical challenges 11
Use of OSS by Chemical Engineering 12
Python 12
Simulation of particle flow and distribution 13
Suitability of OSS for the discipline 14
Value of OSS: lecturer's views 14
Further developments 14
Potential as an open educational resource 15
Possibilities and constraints for making materials available as OER 15
National structures constraining or enabling OER 15
Institutional structures constraining or enabling OER 15
Personal aspects constraining or enabling OER 16
Degrees of openness 16
Key challenges 17
Case study: OSS by Physics 17
Pedagogical problems 17
Use of OSS by Physics 18
VPython 18
Value of OSS: lecturers' views 22
Potential as an open educational resource 22
National structures constraining or enabling OER 22

Institutional structures constraining or enabling OER 22
Departmental structures constraining or enabling OER 22
Agential aspects constraining or enabling OER 22
Degrees of openness 22
Lessons learned from both case studies 24
The strong agential aspect in initiating resource conception, design and development 24
OSS useful resource, but pedagogy still key 24
Free software, but development still costs 24
Small scale grants have seeded ideas, but larger grants needed 24
Way forward 25
Lessons learned 25
Conclusion 25
References 26

List of figures

Figure 1: Two tanks in series containing pulse/tracer particles (shown in red)......................13
Figure 2: 2-D representation of particle 14
Figure 3: 3-D representation illustrating particle age vs. particle composition (Phase 2).......14
Figure 4: Chemical engineering simulation – degrees of technical openness 16
Figure 5: Chemical engineering simulation – degrees of social openness 17
Figure 6: Problem – trajectory of spacecraft 19
Figure 7: Distribution of students' responses 2007 19
Figure 8: VPython solution to spaceship problem 20
Figure 9: VPython simulation of solution for spaceship problem 20
Figure 10: Students' comparison between prediction and result 21
Figure 11: Relationship between students' mathematical and computational models 21
Figure 12: Physics simulations – degrees of technical openness 23
Figure 13: Physics simulations – degrees of social openness 23

Acronyms and abbreviations

CET	Centre for Educational Technology
ICTs	Information and communication technologies
LMS	Learning management system
OER	Open educational resources
OSS	Open source software
SA	Share Alike (Creative Commons license)
UCT	University of Cape Town

Executive summary

This paper describes two case studies where open source software (OSS) has been used to create simulations to assist students in visualising complex processes in university courses. The first case reviews the use of Python to help students visualise the motion of particles or molecules in physical processes used in chemical engineering. The second case reviews the use of VPython to allow students to create their own simulations of abstract concepts in physics.

The aims of this paper are to provide a brief overview of the potential value of OSS for teaching and learning; to discuss the pedagogical needs that led the lecturers to use OSS such as Python and VPython to encourage visualisation of processes or principles; to describe how each of these groups of lecturers and students use OSS in their particular context; to examine how well the simulations seemed to have worked, in order to alert others who may have similar pedagogical needs; and to explore the possibilities and potential constraints of making these resources and processes available as open educational resources (OER) to a wider audience.

In addition, this paper explores the way forward for these two groups of lecturers and students and suggests the possibility of working on a shared project which could be of mutual benefit, and the prospect of making these or some of these resources available as OER.

OER opportunities

These two case studies highlight a number of possibilities for and issues around OER that need to be adopted or resolved respectively:

- The Physics Department has for sometime been hosting its own website for tutorial material and lists of links to other useful resources, indicating an established willingness to share some teaching and learning materials openly.
- Both lecturers and their students are using OSS, so the first hurdle of technical openness (Hodgkinson-Williams and Gray 2008) has been overcome.
- Both lecturers are willing to share their interventions – the actual products (in this case Python or VPython scripts) as well as the processes that they devised to use them in context – first within the university and then beyond.
- What needs to be resolved is the way in which the university understands the copyright on software developed by lecturers and students in the course of their work and study at the university. At the moment the conditions for an academic appointment stipulate that:

  > By accepting appointment you grant to the University a free licence to reproduce, for teaching and examination purposes within this University only, all teaching and examination material you produce in the course of your duties. This licence will be regarded as having lapsed should you publish the material in book form. (Human Resources standard letter of appointment for academics)

  This makes it slightly unclear as to who exactly owns the copyright of software written by lecturers and students.
- In addition, although the current Teaching with Technology grant criteria stipulate the need for recipients to 'be open to collaborative research', they do not stipulate that the resources or processes developed need to be 'open' for the use of others at the university or beyond.

Introduction

In a seminal essay on open source software (OSS), 'The architecture of participation', Tim O'Reilly astutely observes that:

> What really distinguishes open source is not just source, but an 'architecture of participation' that includes low barriers to entry by newcomers …. This architecture of participation allows for a real free market of ideas, in which anyone can put forward a proposed solution to a problem; it becomes adopted, if at all, by acclamation and the organic spread of its usefulness.[1]

This, in essence, is the value of OSS – participation and empowerment. For, as O'Reilly goes on to say: 'Empowerment of individuals is a key part of what makes open source work, since in the end, innovations tend to come from small groups, not from large, structured efforts.'[2]

This paper reviews the use of OSS to encourage visualisation of processes or principles in chemical engineering and physics at university level and how the possibility of the 'architecture of participation' has arisen in these two contexts.

Open source software for teaching and learning

OSS, or particularly free open source software (FOSS), is greatly valued in teaching and learning. Thompson suggests that:

> Open source software will provide new and exciting possibilities for educators. Obviously, the free or low cost availability of open source software has great appeal for educators at all levels ... [and this is an] opportunity provided by open source software for education to adapt software to the needs of their students. In the same way a teacher might adapt a lesson plan to his or her needs; open source software may provide the opportunity to adapt a software program. (Thompson (2002: 101) cited in Baser 2006: para 9)

OSS is obviously used quite extensively in disciplines focused directly on the development of software, such as computer science (O'Hara and Kay 2003). However, OSS is also used quite extensively in physics education (Belloni *et al.* 2008), but less so in chemical engineering education.

Use of open source software in chemical engineering education

With reference to an open process simulator, OPEN CHEMASIM™,[3] Hasse, Bessling and Böttcher (2006) reflect upon the value of being able to trace original code to see how a particular problem was solved and the consequential benefit of using OSS for teaching and learning:

> It is well known, that if the same problem is solved with different codes, even for only moderately complex problems, the solutions often differ outside the numerical uncertainty …. It is therefore highly desirable that, it can be tracked in the code what really was done. This is only possible with Open Source Codes …. More fundamentally, it can be argued that black box simulations are unacceptable for any scientific purpose. One of the most essential requirements of scientific work is repeatability and, more stringent, traceability. Reports on scientific experiments or simulations must put other scientists in a position as to be able to repeat the described experiments or simulations and to trace what has been done in all relevant aspects. (Hasse *et al*. 2006: 255)

---

[1] http://www.oreillynet.com/pub/wlg/3017

[2] http://www.oreillynet.com/pub/wlg/3017

[3] http://chemasim.itt.uni-stuttgart.de

Hasse *et al*. (2006) also emphasise the value of group participation in developing OSS:

> Furthermore, as in principle an unlimited number of people can actively participate in debugging an Open Source Code, these codes will in the long run generally be more reliable than undisclosed codes. (Hasse *et al*. 2006: 255)

Interestingly, the code for OPEN CHEMASIM™ was originally developed as an in-house tool by a commercial company, BASF, and has now been transformed into an open source code for academic use (Hasse *et al*. 2006). Academic institutions are able to use it freely for teaching and research and to distribute it to students for their project work, with the proviso that results from these projects are published unrestrictedly (Hasse *et al*. 2006).

As inherently valuable as the use of OSS in chemical engineering might seem, there seems to be a dearth of reported and sustained use of OSS in chemical engineering education. Although there are other types of computer technology being used in chemical engineering education – such as the use of virtual reality in chemical reaction engineering (Bell and Fogler 1996) or a web-based laboratory (Klein, Hausmanns and Wozny 2005) – there seem to be surprisingly few examples of lecturers or students actually using OSS to model and problem solve chemical engineering principles or processes. This is unlike the situation in physics education, where there has been a much more sustained use of OSS.

Use of open source software in physics education

In contrast to the fairly minimal use of OSS in chemical engineering education, physics education has been using OSS for some time. It is now accepted as part of the physics curriculum – Computational Physics – and has carved out a niche at conferences (for example, see Physics Research and Education: Computation And Computer-Based Instruction[4]); funded projects (for example, see the Open Source Physics Project,[5] the BQLearning Project[6] and the Statistical and Thermal Physics (STP) Curriculum Development Project[7]); and specific articles written about the use of OSS for physics education (for example, see Belloni, Christian and Mason 2008). Lecturers and students are also making use of a wide range of OSS software, including VPython (Salgado 2008), OGRE (Brinton, Shelton & Scoresby 2007), QM Superposition, QM Measurement and Easy Java Simulations[8] (Belloni *et al*. 2008).

This brief overview of the use of OSS in chemical engineering and physics provides the backdrop against which the two case studies at the University of Cape Town (UCT) are explored. Before these cases are presented, the methodological choices that were used in these studies are reviewed.

Background and methodology

Selection of cases

Selection of these case studies was based on their prior application for a Teaching with Technology grant,[9] offered each year by the Centre for Educational Technology (CET) at UCT, and reported success of their implementation of the funds by the fund coordinator in CET. CET, through funding provided by the Mellon Foundation, endeavours to encourage

---

[4] http://www.grc.org/programs.aspx?year=2008&program=physres

[5] http://www.compadre.org/osp/

[6] http://www.bqlearning.org/

[7] http://stp.clarku.edu/

[8] http://fem.um.es/EjsWiki/

[9] http://www.cet.uct.ac.za/collaborate#grants

the use of technology in teaching by awarding relatively small amounts of money to those lecturers who present workable proposals each year. The criteria for the Teaching with Technology grant by implication were embedded in the selection of these case studies. These criteria stipulate that the project should:

- Align with UCT's institutional priorities.
- Comply with UCT's educational technology policy.
- Address specific teaching and learning challenges.
- Focus on teaching and learning rather than research (i.e. it should be used to build students' research capacity, rather than to support staff research projects).
- Be appropriate for a residential university context.
- Be open to collaborative research.

The criteria also specify that the project should *not*:

- Simply involve the development of a website or buying of a DVD to engage students.
- Simply involve moving content from one learning environment to another.
- Replicate existing tools or services.
- Focus on technical aspects such as computer hardware.

In addition to these criteria, the OpeningScholarship case study selection criteria included the possibility of sharing products or processes with others within and beyond the confines of UCT.

In 2006 one of the senior lecturers in Chemical Engineering applied to CET for a Teaching with Technology grant to develop a simulation to help students visualise the physical processes described by various equations used in chemical engineering. His intention was to produce a set of simulations that illustrate graphically the motion of objects representing particles or molecules as they collide with each other and with other entities, according to the physics of the situation. The simulations were intended to be used at first in the classroom and then later as downloadable quizzes and revision exercises for students. His application was successful and he was subsequently awarded R20 000 for the development of a simulation of the activity of particles in a stirred tank reactor for use by third year students in their Reactor Design I & II courses. In 2007 the lecturer again applied for the grant and received a further R20 000 to extend the use of this tool to simulate mass transfer and assist in teaching general topics on fluid flow.

In 2006 one of the associate professors in Physics applied to CET for a Teaching with Technology grant to develop modelling and visualisation in the teaching and learning of physics. He was awarded a grant of R18 000 to develop a library of VPython simulations to support the teaching of the first year physics course.

Case study methodology

This study adopted the case study as a methodology which Robson defines as 'a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon in its real life context using multiple sources of evidence' (1993: 52).

The case study used a range of methods to investigate the use of OSS for visualising complex processes by the Departments of Chemical Engineering and Physics. These included document analysis of the presentations and text resources (field notes, proposals for funding, teaching materials and departmental websites) and follow-up interviews and meetings with both departments.

Initially the researcher interviewed a number of staff members from CET and asked for their opinions on which initiatives were worth pursuing as potential case studies at UCT. Both initiatives in Chemical Engineering and Physics were mentioned frequently. The researcher then attended a presentation by both lecturers as part of the CET's Teaching with

Technology seminars.[10] During these seminars the researcher took copious field notes and was subsequently given access to the electronic presentations. The Physics seminar took place first and the researcher made an appointment to meet with the Physics lecturer at a later date as he was about to leave on sabbatical. In the interim, the researcher made arrangements to meet with the Chemical Engineering lecturer and his programmer. A second meeting was held with the Chemical Engineering lecturer, the programmer, the researcher and the staff training co-ordinator from CET. A draft report with queries tagged using the 'comment feature' was sent to the Chemical Engineering lecturer for comment and clarification. The researcher then held a separate meeting with the OpeningScholarship research director and two lecturers from Physics and one of their postgraduate students. Once it was apparent that there was a possible synergy between these two departments, a combined meeting was held with the Physics and Chemical Engineering lecturers, the researcher and the portfolio manager from the Shuttleworth Foundation. The researcher took additional notes at a subsequent seminar presented by the lecturer from Chemical Engineering and included these in the original report. In addition, CET provided the original proposal documents submitted by both these departments for Teaching with Technology grants which were awarded.

The researcher used the data from these various sources to compile this report which was then sent to the CET Curriculum Co-ordinator (who is responsible for administering the Teaching with Technology grants). The paper was then sent to both parties to confirm its accuracy and comprehensiveness.

Case study: OSS by Chemical Engineering

The lecturer and programmer in Chemical Engineering used the OSS Python to develop simulations to encourage visualisation of key processes in chemical engineering. This was motivated by national challenges as well as pedagogical challenges experienced within the institution.

Contextual and pedagogical challenges

During the interviews, the lecturer explained that South Africa needs to produce more engineers and that the Minister of Education has pledged R48 million to boost the output of engineering graduates at UCT, the University of the Witwatersrand (Wits), the University of KwaZulu-Natal (KZN) and the University of Pretoria.[11] However, he intimated that this brings with it a responsibility to provide quality teaching materials that address the needs of an increasingly diverse group of students.

In his proposal the lecturer explained that his students struggle to visualise the interactions of particles and molecules; the changes in structure of particles (particle breakage or agglomeration) and molecules (reaction); changes in the state of energy (heat transfer); and how these transformations are represented graphically and numerically.

He indicated that students find some of the engineering concepts difficult to understand, that they lack experience of using computers and display inadequate algorithmic thinking. In addition, he observed that many students seem reluctant to ask questions, so he and his colleagues are continually looking for ways in which to help students master engineering concepts in a visually engaging way. They have thus set themselves this strategic goal: 'an order of magnitude improvement in students' understanding of the concepts of chemical engineering' to increase the number of engineers who graduate from UCT.

[10] http://teaching.cet.uct.ac.za/

[11] http://www.thetimes.co.za/Business/BusinessTimes/Article.aspx?id=706105

The lecturer conceived the idea of developing a series of simulations that would help students visualise physical processes to 'extract meaning from the mathematical descriptions'. He believes that this process helps students to 'move from merely attempting to understand the equations to exercising their creativity in accomplishing something practical with this knowledge'.

The lecturer needed the knowledge and skills of a programmer to create the simulation he had initially envisaged. He therefore advertised a programming position on a Linux mailing-list and subsequently engaged the services of a contract programmer.

The lecturer revealed that he received financial support from CET, but no specific pedagogical or technical support as he felt that he and the programmer had a sufficiently clear idea of what they wanted to do. Reflecting on their pedagogical strategy, however, he admits that it was fairly intuitive, and may have benefited from additional pedagogical advice. On the technical side, the programmer did receive some support from Dr Alan Kay (http://vpri.org), who advised him on general issues in the design of educational simulation systems and provided critical insights into visual programming language and notation design.

Use of OSS by Chemical Engineering

The contract programmer uses a generic educational simulation framework which he is authoring himself and uses the OSS Python programming language for interaction.

Python

The Python programming language is a 'dynamic object-oriented programming language that can be used for many kinds of software development'.[12] Its value for educational activities is that it is free, open source and operates on many platforms. Its particular benefit is that it 'offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days … [and encourages] more maintainable code'.[13] Even though Python is a relatively easy introductory programming language 'in which to explore procedural, functional and object-oriented approaches to problem solving' it is sufficiently powerful itself and 'plays well with others (including with multiple operating systems)[14] to be useful beyond the original learning process. It is therefore particularly useful for students who do not need to become computer programmers, but need to understand programming and use it within their own disciplines. With some basic training, students are able to apply the tools to solving problems within their disciplines.

Python has been used by many educators and students and has an established special interest group, the 'EDU-SIG: Python in Education',[15] where interested parties can share ideas and discuss future plans for using Python for teaching and learning. To encourage communication and possible collaboration, there is a dedicated and active mailing list that has been operational since 2000.

For these reasons, the lecturer and programmer adopted Python to create a range of simulations primarily for demonstrations by the lecturer during lectures but also for students to use during practical sessions where they can manipulate the variables in the simulation.

The first development of the simulation involved the simulation of particle flow and distribution.

---

[12] http://www.python.org/

[13] http://www.python.org/

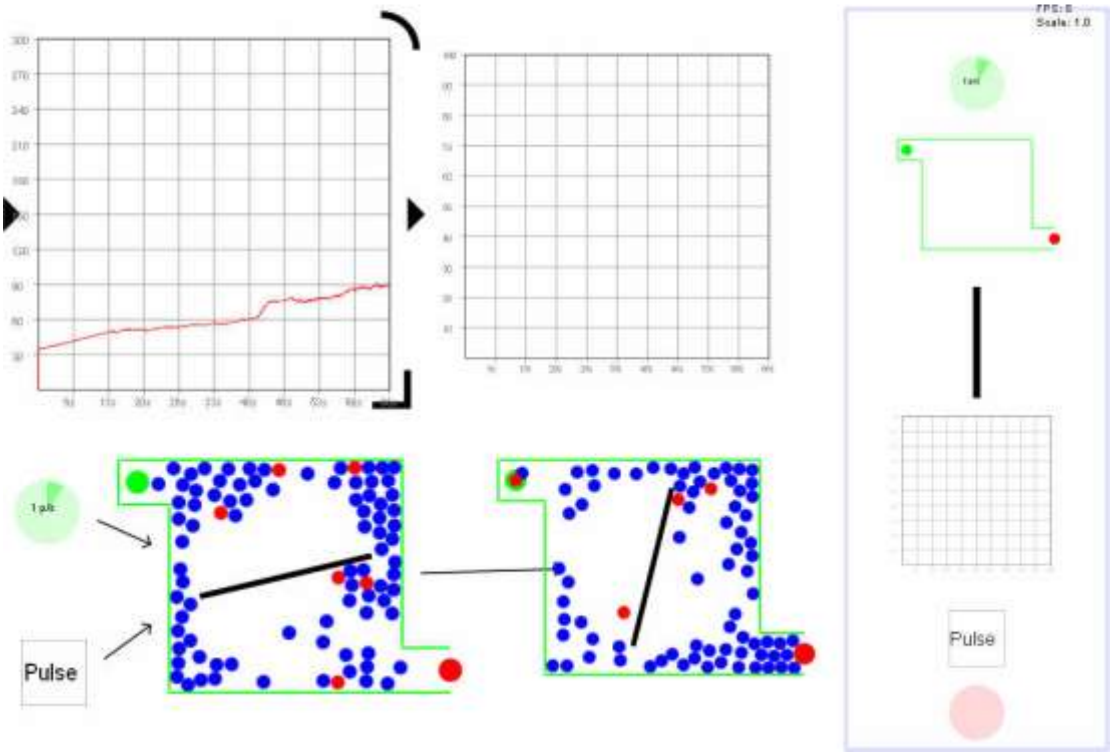
[14] http://www.python.org/

[15] http://www.python.org/community/sigs/current/edu-sig/

Simulation of particle flow and distribution

Error! Reference source not found. provides a snapshot of a simulation where two tanks are connected in series, with particles fed to the first tank. A stirrer in this tank maintains a high degree of agitation. Students can adjust the speed of particle delivery interactively. The particles are in motion and undergo elastic collisions with the walls of the vessel, the agitator, and each other. The two graphs track the number of particles in each tank in real time, allowing students to note the changes. The students can adjust the speed of delivery to the first tank and observe how the distribution of particle mass between the two tanks evolves. In addition, by clicking the pulse button and thus injecting red particles into the reactor, students can observe a simulation of a tracer injection which allows them to follow how particles that entered at a specific time distribute themselves. This is designed to improve students' understanding of the 'mixing' concept in these reactors.

Figure 1: Two tanks in series containing pulse/tracer particles (shown in red)



In September 2008, the lecturer reported a follow-up project on the simulation which was developed in Phase 1 for use in teaching the Residence Time Distribution aspect of the Reactor Design II course (Figure 2**Error! Reference source not found.**). Phase 2 of the project will include the development of 3-D simulations that reveal the influence of diffusion and reaction on the overall reaction rate observed on a particle (Figure 3) and hence on the overall reactor behaviour. This involves applying the routines already written and extending them to the system described above. The lecturer has already written code for the 3-D simulation and to show how particles repulse each other (they repel each other more strongly as they get closer), so that he can simulate diffusion-like effects. He has also included simultaneous reaction (particles change colour), as he plans to illustrate reaction-diffusion problems in this phase. Students will undertake the same types of tests as administered in Phase 1 (2007) and will then analyse the results.

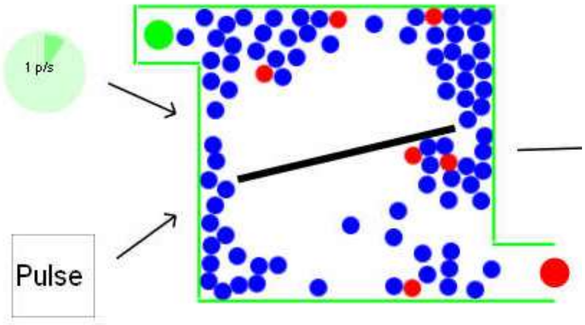


Figure 2: 2-D representation of particle interactions in a reactor (Phase 1)

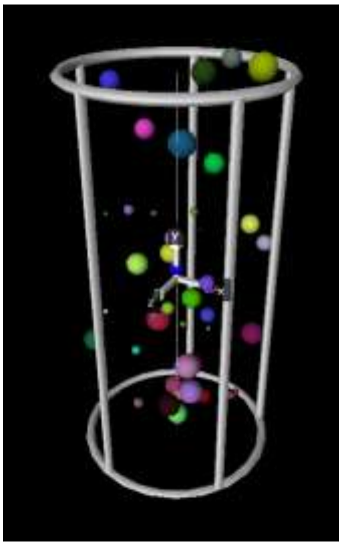


Figure 3: 3-D representation illustrating particle age vs. particle composition (Phase 2)

Suitability of OSS for the discipline

Python is particularly suited to the many South African university students with no previous experience of computer programming. The lecturer envisages that they will not only operate simulations he has created, but also create simulations of their own which they can share with each other electronically.

Additionally, Python is suited to chemical engineering students as it is a relatively easy introductory programming language that can help them model processes.

Value of OSS: lecturer's views

The lecturer asserts that the use of the simulations developed using OSS has improved the students' performance. He undertook a small-scale quasi-experimental evaluation study in which he gave half the class a standard lecture, and the other half the lecture plus the simulation to illustrate the concepts, after which both sets of students completed the same quiz. Students exposed to the simulation achieved approximately 17% higher scores than the group exposed to the standard lecture only. He intends publishing the results of this study.

Further developments

In order to continue the development of the simulations for this course, the lecturer is seeking at least R50 000 in external funding to pay for the programming of these simulations. These include the following extensions to the objects palette:

- Plug flow reactors: it can be shown that almost all real reactors can be modelled as combinations of stirred tank and plug flow reactors. A plug flow object will greatly increase the ability to simulate real reactors.
- Wall objects: at present, a reactor object must be of specific dimensions and the stirrer may only be placed at one location. A wall object would allow them to design reactors of arbitrary shape and thus illustrate the effect of internal geometry on reactor performance.
- Point sources: rather than fixing the location of the feed point of particles to the system, positioning the source of the particles allows them to simulate some of the very complex flow patterns often described in the course.
- Save simulation: they require the ability to save a simulation for subsequent use rather than constructing a simulation from scratch for each demonstration. This would allow

them to prepare complex physical situations off-line for lecture purposes as well as allow students to download and interact with the simulations themselves.

- Reaction: at present, the particles are stable objects; to simulate reaction, they must include a certain probability that when two particles of the same type or of different types collide, the nature of the particles will change. This can be facilitated by colour changes or changes in the particle size.
- Imposed force fields: imposing force fields would allow the simulation of the influence of gravity on a system and even the evolution of temperature and its impact on system performance.

Potential as an open educational resource

In terms of making these materials more publicly available, the lecturer neither supports nor opposes having the simulation offered as an OER, saying that it is 'neither here nor there' to him. However, as an open source programmer, the contract programmer has more clearly defined views on the value of sharing and he strongly supports a more public distribution.

Possibilities and constraints for making materials available as OER

There are, on the one hand, various levels of constraints that hinder lecturers from producing potential OER and, on the other hand, levels of enablements that encourage lecturers to invest time and creativity in developing such materials. These levels can be broadly identified at a national, institutional and personal level.

National structures constraining or enabling OER

In terms of national structures that tend to constrain the lecturer using or producing potential OER, only one slightly implicit constraint alluded to was the time that it took to develop these materials. This reduced the amount of time spent on other activities such as research which is a key requirement stipulated by the national policy and procedures for measuring research output of public higher education institutions. While the lecturer indicated that he had plans to publish his experiences of using ICT-supported teaching strategies in Chemical Engineering, these publications were not a priority.

The only clearly identifiable structural enablement at a national level was the additional funding provided by the South African government for universities in general, but specifically for engineering. However, this general funding has not trickled down to individual lecturer level as yet.

Institutional structures constraining or enabling OER

In terms of institutional structures that tend to constrain the lecturer using or producing potential OER, two clearly identifiable issues emerged: the lack of time and the lack of resources. The time that it took to develop these materials reduced the amount of time spent on other activities such as research which is specified in the Rate for Job policy at UCT which stipulates the minimum peer reviewed outputs per year. While the lecturer has received funding from CET's Teaching with Technology grant on two occasions, this has not been sufficient to complete the set of activities within the simulation. The lecturer is currently seeking further funding from industry and donor foundations to pay for the programmer's specialist skills.

CET's Teaching with Technology grant is an enabling institutional structure that has the potential to encourage lecturers to produce prospective OER. This potential would remain latent unless the lecturers apply for the fund, which in this case the Chemical Engineering lecturer did, being awarded R20 000 in 2006 and again in 2007. The key factor then would seem to be the 'agency' – the 'self-determination, self-expression and strong evaluation' (Alexander 2005: 345) displayed by the lecturer.

Personal aspects constraining or enabling OER

In identifying the personal reasons that seemed to encourage the lecturer and the programmer to modify their teaching practice and develop resources that could possibly be used more widely or maintain their current teaching practice and use of resources, it would seem that the key differentiating level of agency is not necessarily at the level of practice, but tends to be at the level of 'concerns' (Archer 2000; 2003) – the lecturer's and especially the programmer's vision of how they envisaged their simulations being used. The Chemical Engineering lecturer and particularly the programmer displayed a broader vision of the use of their simulations at the commencement of their development. Having these materials distributed beyond the borders of their classroom was not an afterthought, but part of the original learning design. The potential of these simulations can therefore be described along a continuum of openness.

Degrees of openness

Hodgkinson and Gray (2008) describe four attributes of 'openness': social openness – 'the willingness to make materials available beyond the confines of the classroom'; technological openness – 'the presence and use of interoperability standards and functionality'; legal openness – 'a range of flexible licenses'; and financial openness – 'affordability to the user'.

The chemical engineering simulation is strongest on the technical openness (**Error! Reference source not found.**). However, as the programmer has needed to include some of his own code, it might not be as easy to use as a simulation developed only in Python.

Figure 4: Chemical engineering simulation – degrees of technical openness

<table>
<tr><th colspan="3">Technical openness</th></tr>
<tr><td>Proprietary</td><td>Open standards</td><td align="right">Open</td></tr>
<tr><td>proprietary software</td><td></td><td align="right">open source software
• Python</td></tr>
<tr><td colspan="3">◄──────────────────────────────●──►</td></tr>
<tr><td colspan="3">most restrictive
most accommodating</td></tr>
</table>

In terms of social openness (Hodgkinson-Williams and Gray 2008), the chemical engineering simulation is probably situated within the 'lecturer-centred' section, somewhere between 'distributing' and 'coordinating' which reflects how the lecturer is using the simulation beyond his own didactic use (**Error! Reference source not found.**). The lecturer and the programmer have indicated their willingness to open up their simulations to other cognate disciplines.

Figure 5: Chemical engineering simulation – degrees of social openness

Social openness								
Lecturer-centred USE					**Student-centred** CONTRIBUTE		**Student, lecturer and broader community** SHARE	
Traditional	**Distributing**	**Coordinating**	**Organising**	**Managing**	**Participating**	**Contributing**	**Exchanging**	**Collaborating**
Individual lecturer designs, uses & manages own materials for specific context LMS reflects 'silo' model of teaching	Individual lecturer designs, uses & manages own materials for specific context, but shares some Invited access to others' LMS space	More coordinated curriculum design More sharing and reuse Partial access to others' LMS space	Clear aims for working together Group design procedures More organised local materials Common access to the groups' courses in an LMS	Courses designed, developed & managed by multi-disciplinary teams Materials managed centrally	Participate in forums or chats within LMS spaces	Contribute opinions on content and processes in blogs	Sharing content and processes with peers and lecturers using wikis	Collaborating with peers, lecturers and others (e.g. professionals in the field in virtual communities)

Most didactic
Most participative

In terms of legal openness, the lecturer and the programmer have yet to decide what type of license would be most suitable for their simulation, but a GNU license might be worth considering.

In terms of financial openness, the lecturer and programmer have not indicated any direct fee, but have not yet resolved where they would host these simulations if they were able to make them more widely available.

Key challenges

Without doubt the key challenge is to find sufficient funding to continue the development of these simulations and the time to devote to this project. One option is to find a partner in a cognate discipline, for example Physics, where they too make use of OSS and specifically VPython for similar pedagogical needs.

Case study: OSS by Physics

Pedagogical problems

During the interviews, the lecturers explained that the number of students is increasing, but that many students have not had, or have not had sufficient, experience of using computers or of programming. Moreover, as the use of computers is being increasingly incorporated in the curriculum, so the need for students to develop their computer skills increases. The lecturers explained that part of their curriculum change has been the development of Computational Physics. When this course was first started by one of them they made use of Fortran as a programming language, but changed to VPython about seven years ago as it was deemed to be an ideal 3D program. The VPython textbook that was published in 2003 has since been integrated into the course at first-year level. The main purpose at this level is to teach students sufficient programming skills to be able to solve computations. In the first semester of the second year, the focus on computer programming diminishes as it is used less formally, but it becomes more central in the second semester. The lecturers describe the use of computers and programming in the third and fourth years as being 'pervasive across the course'.

A key challenge that they face, like the Chemical Engineering lecturers, is that students struggle to visualise the largely abstract ideas that define their discipline. In his presentation, one of the lecturers explains that: 'The main feature of physics that both defines physics as a discipline is that a small number of abstract ideas (principles and theories) can be applied to a wide range of physical applications, which allow description / explanation / predication of natural phenomena'. It is self-evident that if students struggle to conceptualise and visualise these principles and theories, this will diminish their abilities to describe, explain or predict particular natural phenomena.

The lecturer goes on to describe how students have to make sense of real world phenomena that are 'concrete' and can be experienced or observed and then relate these to physical theories that are 'abstract, a-contextual, external and are manifested in mathematical or linguistic form'. He suggests that physics modelling by experts results in the production of physical models that can help students mediate between the physical theories and reality. It is in the process of understanding current and building new physical models that students make the links between the physical theories and reality. The key to the process of physical modelling is visualisation and this is where the students struggle. Introducing the use of VPython in the classroom was a strategy to address this inability to visualise abstract ideas of physical systems.

Use of OSS by Physics

VPython

VPython is a real time graphics library which is available in a Python API[16] which supports vector computations and can be used to display 3D visualisation of specific objects without the need for complex graphics programming. Sherwood describes VPython as a 'combination of Python (http://python.org), the Numeric module from LLNL (http://www.pfdubois.com/numpy), and the Visual module created by David Scherer, all of which have been under continuous development as open source projects'.[17] He explains that the Visual module includes a set of 3D objects (sphere, cylinder, arrow, etc.), tools for creating other shapes, and support for vector algebra. The 3D renderer runs in a parallel thread, and animations are produced as a side effect of computations, freeing the programmer to concentrate on the physics.[18]

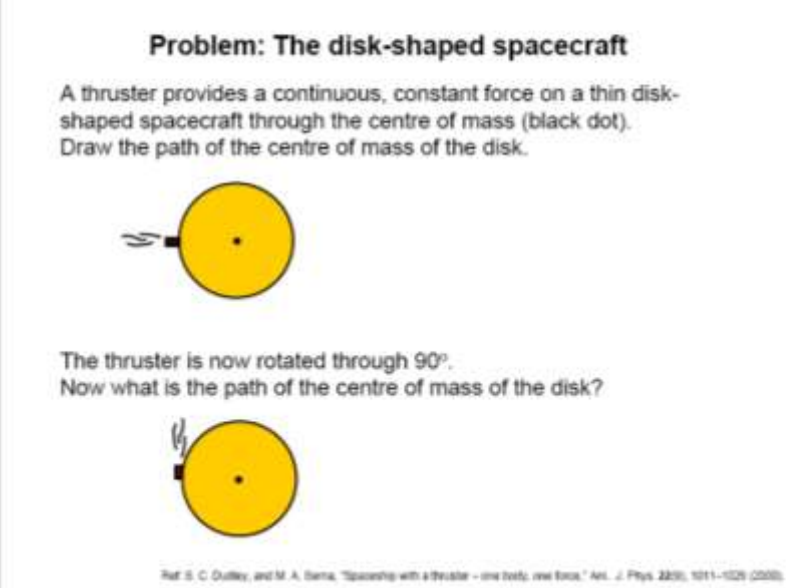
The students are required to use VPython to help them solve a problem. One of the examples in the presentation illustrates the process well. Students are given a scenario – in this case a visual of a disk-shaped spacecraft that has a thruster to propel it – and they are then asked to sketch by hand what would happen if the thruster was rotated through 90° (**Error! Reference source not found.**).

[16] http://vpython.org/

[17] http://flux.aps.org/meetings/YR04/MAR04/baps/abs/S3250.html#SJ5.005

[18] Ibid.

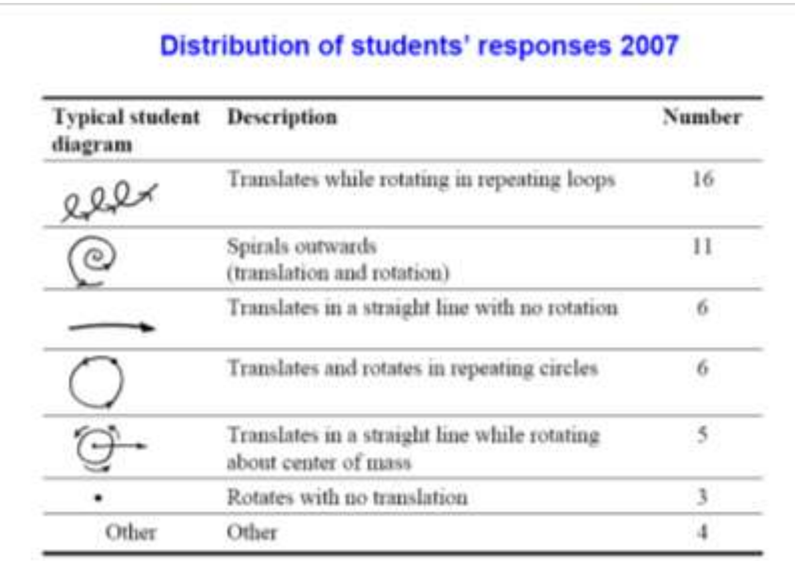
Figure 6: Problem – trajectory of spacecraft



(Source Buffler 2007: 33)

The lecturers have been studying students' predictions of what path the spacecraft would take ( ).

Figure 7: Distribution of students' responses 2007



Distribution of students' responses 2007

Typical student diagram	Description	Number
	Translates while rotating in repeating loops	16
	Spirals outwards (translation and rotation)	11
	Translates in a straight line with no rotation	6
	Translates and rotates in repeating circles	6
	Translates in a straight line while rotating about center of mass	5
•	Rotates with no translation	3
Other	Other	4

(Source Buffler 2007: 35)

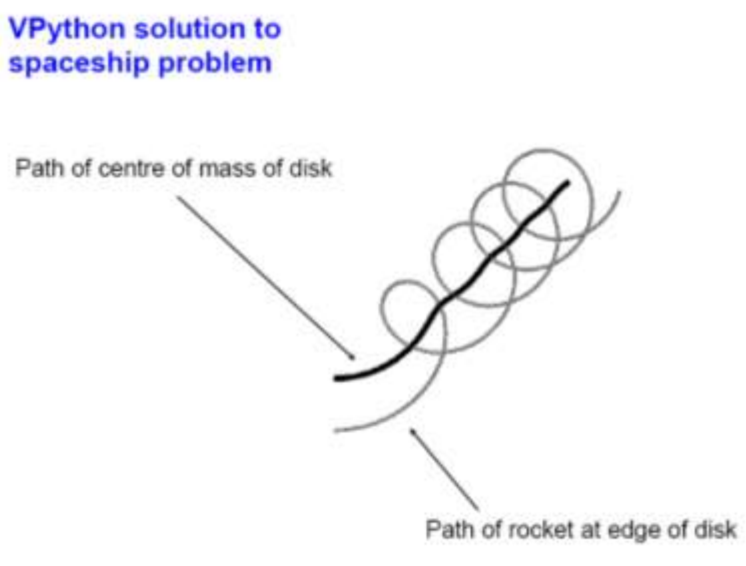
The students are then required to use VPython to check their prediction by programming in the applicable variables ( ).

Figure 8: VPython solution to spaceship problem

(Source Buffler 2007: 36)

Once the students have correctly programmed the scenario in VPython, they are able to visualise what path the space craft would take ( ).

Figure 9: VPython simulation of solution for spaceship problem



(Source Buffler 2007: 34)

The key learning moment is the comparison between the students' initial predictions and the VPython simulation.

To establish to what extent the visualisation process is able to help students better understand the problem, lecturers require the students to compare and explain their original prediction in relation to the VPython simulation ( ).

Figure 10: Students' comparison between prediction and result

Category	Number
No comparison made	12
Comparison made, but with no explanation	14
Comparison made, supported by non-physical explanation	14
Comparison made, supported by explanation based on physical model	11

(Source Buffler 2007: 38)

The lecturers were also able to assess the relationship between the students' mathematical and computational models ( ).

Figure 11: Relationship between students' mathematical and computational models

Category	Number
Both mathematical and computational models incorrect	9
Mathematical model incorrect, computational model correct	8
Mathematical model correct, computational model incorrect	14
Both mathematical and computational models correct	20

(Source Buffler 2007: 37)

The most interesting result of the exercise is that 'none of the students could correctly predict the behaviour of the spacecraft, although many could write down the mathematical model of the problem' (Buffler 2007: 39). The lecturer notes that by the end of the VPython session, 'nearly all students had working codes', but is quick to add that this 'does not imply that an appropriate mathematical model was in place' (Buffler 2007: 39). What was of more value was that by the 'end of session, 11 of the 51 students could offer plausible explanation for the observed behaviour of the spacecraft, although 14 more did attempt some form of explanation' (Buffler 2007: 39).

In reflecting on the strategy of using VPython to assist students in visualising physical models, the lecturer comments quite honestly that their 'attempts to have the students reflect upon the process, interpret their outputs in relation to their predications, and offer a physical explanation for the observed behaviour of the system were only partially successful' (Buffler 2007: 39). He remarks that 'many students were not able to progress beyond merely describing what was displayed in the visual output of their programs' (Buffler 2007: 41) and proposes that the pedagogical strategy be reviewed. He recommends that students have their mathematical models reviewed by tutors before they start programming in VPython so that tutors can alert students to the underlying mathematical model and 'not simply provide technical answers in the syntax of the programming language' (Buffler 2007: 41).

Value of OSS: lecturers' views

According to the lecturers, the first year students are very responsive to being taught VPython to solve computations.

Some of the concerns that they raise include the lack of time to develop additional VPython scripts as well as an online resource to assist the students in learning VPython. These scripts could be written by senior students.

Potential as an open educational resource

The Physics Department has for some time been hosting its own website for tutorial material and lists of links to other useful resources, indicating an established willingness to share some teaching and learning materials openly.

National structures constraining or enabling OER

The Physics lecturers, like the Chemical Engineering lecturer, noted that national structures tend to constrain lecturers using or producing potential OER in terms of available time. However, the Physics Department has involved a PhD student in this study and one of the Physics lecturers has published some of his findings on the use of VPython. This meets at least two of the demands made upon academics by the Department of Education.

Institutional structures constraining or enabling OER

While time is still a constraining factor, the lecturers have made good use of the funds provided by the Teaching with Technology grant and have participated in a number of Teaching with Technology seminars, in this way sharing their practice with others.

Departmental structures constraining or enabling OER

All the Physics scripts and worksheet-based tutorials are hosted online, either on the PHY1004W course website (www.phy.uct.ac.za/courses/phy1004w) or on the more general repository of VPython programs (www.phy.uct.ac.za/demonline) provided by the Physics Department.

Agential aspects constraining or enabling OER

Once again, personal motivation and willingness to share seem to be the overriding features of the Physics Department's use of OSS; this bodes well for the sharing of existing materials and future resources. The lecturers anticipated the sharing of their materials when they began developing a library of VPython simulations to support the teaching of the first year Physics course. These simulations and the pedagogical processes can be mapped along the four attributes of openness (Hodgkinson-Williams and Gray 2008).

Degrees of openness

As in the Chemical Engineering project, simulation is strongest in terms of technical openness (**Error! Reference source not found.**), as the Physics Department are using OSS and producing 'scripts' that can easily be shared.

Figure 12: Physics simulations – degrees of technical openness

Technical openness		
Proprietary	**Open standards**	**Open**
proprietary software		open source software • VPython
most restrictive most accommodating		

In terms of social openness (Hodgkinson-Williams and Gray 2008), the Physics simulations are probably situated in the 'student-centred' section, somewhere between 'participating' and 'contributing', which reflects how the lecturers are using the simulations to encourage students to engage directly in their development (**Error! Reference source not found.**). The lecturers have indicated their willingness to share their library of simulations with other cognate disciplines.

Figure 13: Physics simulations – degrees of social openness

Social Openness								
Lecturer-centred USE					**Student-centred** CONTRIBUTE		**Student, lecturer and broader community** SHARE	
Traditional	**Distributing**	**Coordinating**	**Organising**	**Managing**	**Participating**	**Contributing**	**Exchanging**	**Collaborating**
Individual lecturer designs, uses & manages own materials for specific context LMS reflects 'silo' model of teaching	Individual lecturer designs, uses & manages own materials for specific context, but shares some Invited access to others' LMS space	More coordinated curriculum design More sharing and reuse Partial access to others' LMS space	Clear aims for working together Group design procedures More organised local materials Common access to the groups' courses in an LMS	Courses designed, developed & managed by multi-disciplinary teams Materials managed centrally	Participate in forums or chats within LMs spaces	Contribute opinions on content and processes in blogs	Sharing content and processes with peers and lecturers using wikis	Collaborating with peers, lecturers and others (e.g. professionals in the field in virtual communities)
Most didactic Most participative								

In terms of legal openness, the lecturers and the programmer have yet to decide what type of license would be most suitable for their simulations, but a GNU or a Creative Commons license is worth considering. What needs to be resolved is the way in which the university understands the copyright on software developed by lecturers and students in the course of their work and study at the university. Although the current Teaching with Technology grant criteria stipulate the need for recipients to 'be open to collaborative research', they do not stipulate that the resources or processes developed need to be 'open' for the use of others at the university or beyond.

In terms of financial openness, the lecturer and programmer have not indicated any direct fee, but have not yet resolved where they would host these simulations if they were able to make them more widely available.

Lessons learned from both case studies

There are a number of useful lessons to be learned from these particular studies. These include: the importance of the personal motivation of the lecturers; the value of OSS as a useful resource, but the primacy of pedagogy; the acknowledgement that, although the software is free, there are development costs; and the enablement of seed funding from small grants, but the need for more substantial grants to undertake sufficient development to create a critical mass of resources.

The strong agential aspect in initiating resource conception, design and development

Both cases point to the key role of the individual lecturers in envisaging the original idea and finding ways – both in term of time and money – to design and develop these materials in accordance with their perception of what their students need. This points to a far deeper 'concern' (Archer 2003), not merely at the level of their everyday teaching practice.

OSS useful resource, but pedagogy still key

The small quasi-experimental research study undertaken by the Chemical Engineering lecturer indicated that there was a 17% improvement in marks due to the use of the chemical engineering simulation. Similarly, the investigation undertaken by the Physics lecturers indicated that there was a 6% improvement, as initially no students predicted the correct trajectory and in the end 11 out of 51 did. It is, however, quite difficult to confirm that this percentage change was actually due to the use of the OSS intervention alone as there are so many other variables in the teaching and learning process that may affect the students' performance (e.g. their prior knowledge, level of motivation, and perception of the lecturers and/or tutors). However, the lecturers' own perceptions of what could have made the intervention better point to pedagogical changes. The Chemical Engineering students indicated that they would like more hands-on experience and not just the demonstration, while the lecturers in Physics recommended that in the future tutors should first assess the students' mathematical models and assist with difficulties in conceptualisation at this point, before requiring students to write the program to illustrate the mathematical model.

Free software, but development still costs

As aptly described by the Free Software Foundation, 'free software is a matter of liberty, not price'.[19] In both the Chemical Engineering and the Physics departments students and lecturers have been able to use, adapt and distribute the software freely. However, there are associated costs with developing discipline-specific scripts for demonstration purposes that require time investment by the lecturers or the acquisition of funds to pay a programmer, as in the case of Chemical Engineering, or for a senior student in the case of Physics. So, while students have had the freedom to run, copy, distribute, study, change and improve Python or VPython, it has required and will continue to require additional funding to maintain and develop the use of OSS in the curriculum.

Small scale grants have seeded ideas, but larger grants needed

Both departments applied for and were awarded Teaching with Technology grants co-ordinated by CET and sponsored by the Mellon Foundation. Both departments presented carefully thought-through grant proposals which outlined the intended projects and provided definable outcomes which have acted as prototypes of the next projects that the departments would like to undertake. They will need to be creative in developing programs/scripts that will attract sufficient funding for the departments to develop a 'critical mass' of materials.

[19] http://www.gnu.org/philosophy/free-sw.html

Way forward

What is clear is that materials are not going to be developed without some planning, co-ordination and funding. The key element is to find creative ways of developing programs/scripts that will attract sufficient funding for the departments to develop a 'critical mass' of materials, which include:

- Working with other cognate departments within the university to cut development time and share resources (e.g. Chemical Engineering, Physics, Mathematics) – possibly with postgraduate students leading the way.
- Working with the Western Cape Education Department and/or other donor-funded projects such as Siyavula, to develop materials suitable for high schools and first year university level (particularly for Physics).
- Working with Chemical Engineering and Physics departments in other universities in South Africa.
- Working with Chemical Engineering and Physics departments in other universities around the world.

Funding sponsorship is a key issue, but only when a carefully thought-through plan has been developed. Ideas include:

- Re-applying for the CET's Teaching with Technology grant to enable funding of postgraduate students to continue the current, albeit small scale development.
- Applying to the donor funders who explicitly support OER: for example, the Shuttleworth Foundation in South Africa; and the Hewlett Foundation and the Open Society Institute internationally.
- Applying for corporate donor funding from associated industry, e.g. Sasol and Engen.

The challenge is to find a champion who will spearhead this process of interdisciplinary, intra- or inter-university collaboration.

Lessons learned

There are a number of useful lessons to be learned from these two studies:

Both lecturers have responded to their students' difficulties with symbolic notation by using simulations to help them visualise complex and/or abstract concepts and processes, which indicates the benefit of lecturers devising specific pedagogical interventions to meet the students' needs.

Both lecturers have applied for and received funding from the university to develop their interventions, indicating the necessity of having a source of funding for teaching and learning interventions. This does, however, require the willingness of lecturers to submit formal funding proposals, engage in or supervise the process of developing these interventions and report on their progress on a regular basis.

Both lecturers have expressed an interest in possibly working on a shared project with other cognate departments where students seem to be experiencing similar conceptual difficulties.

Conclusion

At face value, the use of open source software to create simulations to assist students in visualising complex processes within university courses seems both useful and cost-effective; there are, however, embedded costs in developing a critical mass of resources that can be shared both within the university and beyond. We also need to recognise that the pedagogical strategies are crucial to the optimal use of these resources.

References

Alexander, HA (2005) Human agency and the curriculum. Theory and Research in Education, 3(3): 343–369.

Archer, MS (2000) Being human: the problem of agency. Cambridge: Cambridge University Press.

Archer, MS (2003) Structure, Agency and the Internal Conversation, Cambridge: Cambridge University Press.

Baser, M (2006) Promoting conceptual change through active learning using open source software for physics simulations. Australasian Journal of Educational Technology, 22(3): 336–354. http://www.ascilite.org.au/ajet/ajet22/baser.html [accessed 19 October 2008].

Bell, JT and Fogler, HS (1996) Vicher: A Virtual Reality Based Educational Module for Chemical Reaction Engineering. Reprinted from Computer Applications in Engineering Education, 4(4) October 1996. http://www.cs.uic.edu/~jbell/Professional/Papers/VICHPAPR_Complete.PDF [accessed 19 October 2008].

Belloni, M, Christian, W and Mason, B (2008) Open Source and Open Access Resources for Quantum Physics Education. [online] http://www.compadre.org/quantum/document/ServeFile.cfm?ID=6954&DocID=320

Brinton, M, Shelton, B and Scoresby, J (2007) Exploratory Physics Learning Using Open Source Technologies, Open Education Conference, September, 2007 http://www.51weeks.com/presentation_files/0000/0076/OpenEd2007-Brinton.ppt [accessed 19 October 2008].

Buffler, A, Lubben, F, Pillay, S, Ibrahim, B (2007) Modelling and visualization in the teaching and learning of physics. Centre for Education Colloquium, University of Cape Town 11 October.

Hasse, H, Bessling, B and Böttcher, R (2006) OPEN CHEMASIM™: Breaking Paradigms in Process Simulation. 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering. http://www.nt.ntnu.no/users/skoge/prost/proceedings/escape16-pse2006/Part%20A/Volume%2021A/N52257-Topic1/Topic1-%20Oral/1286.pdf [accessed 19 October 2008].

Hodgkinson-Williams, CA and Gray, E (2008) Degrees of Openness: The emergence of Open Educational Resources at the University of Cape Town. Emerge Conference, Cape Town, June 26-27.

Klein, A; Hausmanns, C and Wozny G. (2005) Web-based remote experiments with a real technical plant in chemical engineering education. http://www.nt.ntnu.no/users/skoge/prost/proceedings/aiche-2005/non-topical/Non%20topical/papers/552e.pdf [accessed 19 October 2008].

O'Hara, KJ and Kay, JS (2003) Open source software and computer science education, Journal of Computing Sciences in Colleges, 18(3): 1–7.

Robson, C (1993) Real world research. Oxford: Blackwell.

Salgado, R (2008) VPython for Teaching Physics: Visualizations for Mechanics, Electromagnetism, and Relativity. Presented at the Physics Research and Education: Computation And Computer-Based Instruction Conference, Bryant University, Smithfield, June 8-13.